



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #86 - June 2014



Photo: Sam Dal Monte (Flickr.com)



HEARTBLEED & TRUECRYPT WHAT ON EARTH IS GOING ON?

Full Circle Magazine is neither affiliated with, nor endorsed by, Canonical Ltd.

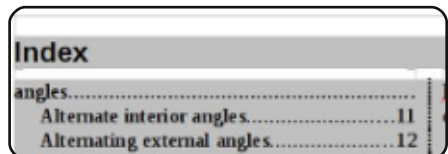


HowTo



Python

p.10



LibreOffice

p.17



GRUB2 & Multibooting

p.21



Blender

p.23



Inkscape

p.25

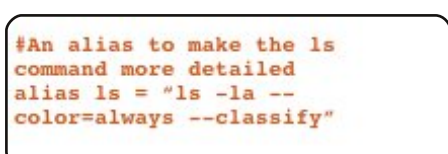


Graphics



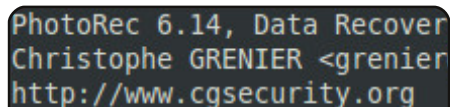
Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY



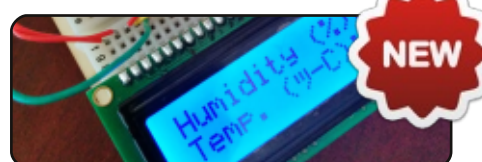
Command & Conquer

p.06



Linux Labs

p.32



Arduino

p.28



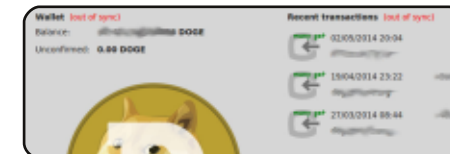
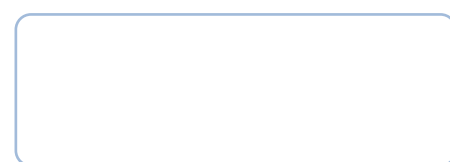
Review: Ubuntu 14.04

p.36



Linux News

p.04



CryptoCurrency

p.39



Letters

p.40



Tuxidermy

p.42



Q&A

p.43



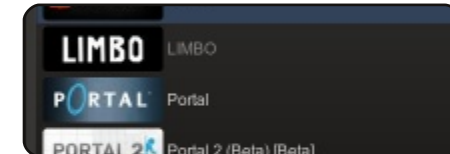
Security

p.45



Open Source Design

p.47



Ubuntu Games

p.49



The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: you must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('Full Circle Magazine') and the URL www.fullcirclemagazine.org (but not attribute the article(s) in any way that suggests that they endorse you or your use of the work). If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

Full Circle magazine is entirely independent of Canonical, the sponsor of the Ubuntu projects, and the views and opinions in the magazine should in no way be assumed to have Canonical endorsement.



WELCOME TO ANOTHER ISSUE OF FULL CIRCLE.

It's a full house again this month. There's Python, LibreOffice, and the second part of the GRUB boot menu and multibooting series. Inkscape and Blender are here and, as you can see from the cover, we've a special Heartbleed/TrueCrypt report from Kevin O'Brien (with addendum by Michael Kennedy). The TrueCrypt thing certainly seems to be quite controversial bordering on conspiratorial. My Arduino tinkering continues with the addition of a humidity sensor to the LCD screen. It's the easiest thermometer ever! I've also written a quick Linux Labs piece (Charles is off on holiday). It's actually from several months ago and after the crash of which we shall not speak. I did it for morbid curiosity more than anything else, but it was interesting to see what files I could recover after a double repartitioning and double Mint reinstall.

Unfortunately, this month we must bid farewell to David Rhys (Ubuntu Games) and Copil (Ask The New Guy) who are moving on to pastures new. I wish them all the best. It's been a pleasure working with them. If you can write something each month for FCM feel free to drop me an email with a few lines explaining what you'd like to write about. Think ahead though, and think of maybe ten, or twelve, subjects. That way you won't write two articles then run out of material.

All the best, and keep in touch!

Ronnie

ronnie@fullcirclemagazine.org



This magazine was created using :



Full Circle Podcast

Released monthly, each episode covers all the latest Ubuntu news, opinions, reviews, interviews and listener feedback. The Side-Pod is a new addition, it's an extra (irregular) short-form podcast which is intended to be a branch of the main podcast. It's somewhere to put all the general technology and non-Ubuntu stuff that doesn't fit in the main podcast.

Hosts:

- Les Pounder
- Tony Hughes
- Jon Chamberlain
- Oliver Clark



<http://fullcirclemagazine.org>



AUDIO MP3



AUDIO OGG

Download



STEAM HITS THE BIG 500 FOR LINUX GAMES

500 Linux compatible games are now on Steam, which is a pretty great number to point anyone at. No longer will people keep stating "but Linux has no games." Sadly, they will say Linux has very little AAA games which is true, but this will be a gradual build up, of course.

If the Steam Machines/SteamOS are successful, we should see the number rise even quicker.

Source:
<http://www.gamingonlinux.com/articles/steam-hits-the-big-500-for-linux-games.3849>

Submitted by: **Liamdawe**

PLAYCANVAS 3D WebGL GAME ENGINE NOW OPEN SOURCE

PlayCanvas is the "world's easiest to use WebGL Game Engine." It's free, open source and backed by "amazing" developer tools. Well now, isn't this interesting.

It has been worked on for the past three years and is now under the MIT license, so you can pretty much do with it as you see fit.

Now, it is just a matter of getting browser developers to make the experience less annoying to game inside a browser, and the OS you use becomes a little less relevant.

Source:
<http://www.gamingonlinux.com/articles/playcanvas-3d-webgl-game-engine-now-open-source.3843>

Submitted by: **Liamdawe**

UBUNTU MATE FLAVOR COULD ARRIVE SOON, PROTOTYPE LOOKS GREAT ALREADY

Ever since the introduction of Unity, some of the Ubuntu users have been pining after GNOME 2, the desktop environment in use until Ubuntu 11.04 arrived. It had a lot of fans, and a part of the Linux community is still hoping that the good days will return.

Martin Wimpress, a MATE Desktop team member, took it upon himself to make an Ubuntu prototype featuring MATE, which greatly resembles the old style used by Canonical until 2011.

This is just preliminary work and it's more like an experiment than anything else, but the developer had help from Canonical's Alan Pope. He left a message saying that something interesting might come out of this: "there's something cooking and it smells delicious. Thanks to Alan Pope for the help."

Source:
<http://news.softpedia.com/news/Ubuntu-MATE-Flavor-Could-Arrive-Soon-445509.shtml>

Submitted by: **Silviu Stahie**

LINUX KERNEL 3.10.41 LTS IS AVAILABLE FOR DOWNLOAD

The latest version of this branch of the stable Linux kernel, 3.10.41, has been announced by Greg Kroah-Hartman and comes with quite a few changes and fixes.

The amount of changes and enhancements for this branch of the Linux kernel is rather large and the developers have added numerous drivers and other improvements. This is a LTS release and it's likely that it will be updated for a long time.

If you are using any of the versions released until now in the Linux kernel 3.10.x branch, it's

recommended that you update to this current version.

Source:

<http://news.softpedia.com/news/Linux-Kernel-3-10-41-LTS-Is-Available-for-Download-445652.shtml>

Submitted by: **Silviu Stahie**

LIBREOFFICE 4.3 BETA 2 IS NOW AVAILABLE FOR TESTING

The developers from The Document Foundation have released a new build in the LibreOffice 4.3 Beta branch, bringing even more changes than the latest update in the series. It looks like 4.3 will be quite interesting, but it's going to take a while until it's released.

Some fixes, according to the changelog, are the upper margin of the multi-page floating table for WW8 import has been fixed, the wrong text position in a grouped list has been corrected, the direct formatting for numbering in .DOCX is now handled correctly, and numerous other fixes have been

implemented.

Remember that this is a development version and it should NOT be installed on production machines. It is intended for testing purposes only.

Source:

<http://news.softpedia.com/news/LibreOffice-4-3-Beta-2-Is-Now-Available-for-Testing-445664.shtml>

Submitted by: **Silviu Stahie**

LINUX MINT 17 WITH CINNAMON DESKTOP KEEPS FOCUS ON EASE OF USE

Linux Mint is among the most popular Linux desktop distributions in use today, thanks in large part to its core focus on improving the desktop experience for users. It's a focus that has been in place for Linux Mint since day one. When Clement Lefebvre developed Linux Mint in 2006, he did so with the goal of creating a user-friendly desktop version of Linux. Linux Mint is based on Ubuntu Linux, adding new desktop, setting and configuration

elements. The latest version of Linux Mint, version 17 (code-named Qiana), is based on the recent Ubuntu 14.04 "Trusty Tahr" release, which is what is known as a Long Term Support (LTS) release.

Source:

<http://www.eweek.com/enterprise-apps/slideshows/linux-mint-17-with-cinnamon-desktop-keeps-focus-on-ease-of-use.html>

Submitted by: **Peter Odigie**





Last month we covered a series of examples when it came to using Git in combination with Github. Within this article, I asked if there was any interest in an article on hosting/creating your own git repository host. Turns out... there is. As such, we'll be dedicating this month's article to creating and hosting your own git repositories, as well as discussing how to manage specific branches (such as: cloning a single branch from a repository, merging branches, creating a new one, etc.).

GIT SERVER

The easiest way to configure a git server would be to simply install git on the server, and configure an SSH server. As this was the method I used, it will be what we focus on this month, and I will assume you have a working SSH server installed on the remote machine already. If you prefer to try this out on a local machine and simply copy repositories back and forth, you can simply use normal file paths instead of the SSH format.

CREATING A NEW REPOSITORY

Assuming this is a remote host, you'll need to SSH into the machine (using the same user you plan on utilizing as the git user). Once you've done that, you're ready to create the repository with the following:

```
git --bare init <folder>.git
```

If the folder doesn't exist, it will be created. I would recommend keeping the server organized by placing all git repositories within a sub-folder of the user's home folder. Something like /home/gituser/git-repos/. As for the command itself: --bare tells git to initialize the repository without a separate .git folder. It seems the standard practice is to use a bare repository for shared repositories (i.e. ones you want people to clone/push/pull/fetch). If you're creating this repository on a local machine and plan to only occasionally access the folder via another machine, you may be fine

without the --bare switch. Otherwise you shouldn't run into issues one way or the other.

ADDING FILES TO THE REPOSITORY

Regardless of whether you initialized the repository in an empty folder, or a folder you've already filled, nothing will be added to the actual repository by default. You'll need to run:

```
git add .
```

Before anything is added. Once you've added it, you'll also need to commit the changes with:

```
git commit -m "Message"
```

Replace message with your actual commit message. Alternatively, you can do both things at once with:

```
git commit -a -m "Message"
```

The -a switch tells git to add and commit everything in the directory. As such, if you want to add only

some files, you'll want to either create a .gitignore, or add the files separately using git add.

Now that the repository is created and contains content, it's time to clone it to a new machine.

CLONING A GIT REPOSITORY VIA SSH

Assumptions:

- You're using the normal ssh port (21)
- Your username is gituser
- The server's domain is git.example.com
- The path is /home/gituser/git-repos
- The repository itself is called cc-example.git

Based on these assumptions, the git clone command would look like this:

```
git clone  
ssh://gituser@git.example.com  
:21/home/gituser/git-  
repos/cc-example.git
```

If you created the repository without the .git ending (or created

a repository of an older directory), you'll just need to adapt the path to reflect this (so it would read "cc-example" at the end of our example). Assuming you haven't set up SSH with keyfiles, you'll be prompted to accept the fingerprint, and to enter your password.

The SSH format for git is as follows:

```
ssh://<user>@<host>:[port]<absolute path>
```

Replace <user> with the actual SSH username, <host> with the actual IP/Domain/Hostname, [port] with the port you're using (you can leave this out if you're using the standard port), and the <absolute path> must always be absolute – meaning it starts from the root filesystem directory:

- Correct: /home/gituser/git-repos/cc-example.git
- Wrong: ~/git-repos/cc-example.git, git-repos/cc-example.git, etc.

If you're using the standard port, you can shorten the format a little by instead writing it like this:

```
git clone  
<user>@<host>:<absolute path>
```

However, it doesn't take much more effort to use the complete format, which may also help reduce errors when working with non-standard port values.

Once you've cloned the repository, you can git add, git commit, and then (in order to synchronize the changes) use git push. The format of this command (like last month) is:

```
git push <remote-target>  
<branch>
```

Typically, the <remote-target> will be origin, and the <branch> will be master. So, a typical command could be:

```
git push origin master
```

If you run into an error (such as the remote origin not being defined), you'll need to add the target to your repository. To do this, change directory to the repository, and then run:

```
git remote add origin  
ssh://gituser@git.example.com:  
21/home/gituser/git-repos/cc-example.git
```

This will define a remote target called origin in the repository, and

use the URL you supplied. This shouldn't typically be required (at least, in my testing I never needed to define a host like this). You can also use it to define multiple remote targets, in case you have various backup servers – though that could easily end up being very complex.

BRANCHES

The reader who contacted me also wished for some information on creating, merging and cloning specific branches in a repository. Anyone planning to get into serious development with git will want to learn about branches, in order to keep the development snapshot separate from the stable.

CREATING A NEW BRANCH

Make sure your current working directory (cwd) is that of your repository, and then type the following command:

```
git checkout -b <branch>
```

This will create a new branch, called <branch>. It's technically short-hand for the following two commands:

```
git branch <branch>  
git checkout <branch>
```

As you can see, the short-hand is a lot less repetition. These two steps are also required only if you're creating branches – changing between branches is as simple as writing:

```
git checkout <branch>
```

Once you've changed to the branch you want to work on, continue working as you normally would (edit files, add them, and commit them). However, there is an important change to the push step:

```
git push origin <branch>
```

To push your new branch to the remote host "origin", you need to make sure you supply the correct branch name. Typically commands use master as a default value here, but that's true only if you're updating the master (i.e. stable) branch.

Assuming you've completed development in the development branch, and are ready to merge it back into the stable (master) branch, then you would need to do

the following:

```
git checkout master
```

This command switches you back to the stable branch – when merging you need to have checked out the target branch. Then merge the branches with:

```
git merge <branch>
```

Make sure you enter the actual branch name. This type of merge uses the typical git approach to conflicts – if it can't automatically resolve the conflict, it will instead mark the changes in the file within the repository, and you need to resolve it manually, then re-add and commit the changes. See last month's article for more detail. If you typically develop in a linear way (i.e. the stable branch always points to an older point in the timeline, and the development branch is more recent), it shouldn't be an issue. However, if you have concurrent branches (i.e. you develop onwards normally after the most recent stable release, but also branch off into a mobile development from the same snapshot), it may cause some conflicts when merging.

DELETING A BRANCH

Deleting an old branch locally is as simple as running:

```
git branch -D <branch>
```

However, if you want to remove it from the remote host as well, you'll need to do one of the following:

```
git push origin :<branch>
```

```
git push origin --delete  
<branch>
```

The difference is that the top command is supported in versions of git as of 1.5.0, and the second one is supported only as of 1.7.0.

RENAMING A BRANCH

If you want to rename a branch locally (i.e. from development to dev):

```
git branch -m <old> <new>
```

So, in the example:

```
git branch -m development dev
```

Also, if you want to rename the current branch, you can omit the <old> part of the command, i.e. git

branch -m dev.

Renaming a local branch when pushing it to the remote server

Say, for example, you have a branch called testing on your copy of the repository. Problem is – someone beat you to it and created a branch called testing with different changes from your own. You could, naturally, rename your branch first, and then push it. Or else tell the remote target what to rename the branch to when pushing it with the following command:

```
git push origin  
<local>:<remote>
```

So, in our example:

```
git push origin  
testing:mobile
```

This will take our local testing repository and upload it to the server, while renaming that branch mobile. This may also help people to understand the deletion command in git 1.5.0: you're essentially pushing a NULL repository (i.e. one that doesn't exist) to the remote branch, which deletes it.

Checking out a specific branch

This was the last question posed in the email I received. I assumed he meant literally cloning a single branch and ignoring all the rest. This is a slightly more complicated task, and it is outlined below:

```
mkdir <folder>
```

```
cd <folder>
```

```
git --bare init
```

Alternatively, just run git --bare init <folder>

```
git remote add origin  
ssh://gituser@git.example.com  
:21/home/gituser/git-  
repos/cc-example.git
```

This is required in order to link the remote repository with the new local repository you just created, that will end up containing only the branch you want.

```
git fetch origin  
<branch>:refs/remotes/origin/  
<branch>
```

There are a few things to note about this command: If your remote target is something other than origin, change both occurrences of origin in the command. Also, replace <branch> with the name of the branch. Do

not change the part that reads `"/refs/remotes/"`. This essentially prepares the download of the specific branch you want from the repository.

```
git checkout -b <branch> --
track origin/<branch>
```

This now creates the branch in your local repository, and then links it with the branch from the remote target – effectively creating a repository containing only that branch.

Note: If you don't mind downloading all the existing branches, and simply want git to default to a different branch (i.e. if you plan to merge branches later), you can do it much easier with:

```
git clone
ssh://gituser@git.example.com
:21/home/gituser/git-
repos/cc-example.git -b
<branch>
```

This clones the repository as normal (including all branches), and then switches the default branch (i.e. master) to the branch you specify (i.e. testing). This would generally be my preference, as opposed to the complicated series of steps listed above. Most likely

you'll eventually need access to at least some of the other branches, and this makes it relatively painless to switch between them.

Hopefully this has helped explain some of the intricacies of managing git branches and servers. If you have any follow-up questions, or ran into any issues with the examples in the article, feel free to email me at lswest34+fc@gmail.com. You're also very welcome to email me with requests for articles, or if you want to offer your 2 cents on any of the steps outlined here.



Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.

PYTHON SPECIAL EDITIONS:



<http://fullcirclemagazine.org/issue-py01/>



<http://fullcirclemagazine.org/issue-py02/>



<http://fullcirclemagazine.org/python-special-edition-issue-three/>



<http://fullcirclemagazine.org/python-special-edition-volume-four/>



<http://fullcirclemagazine.org/python-special-edition-volume-five/>



<http://fullcirclemagazine.org/python-special-edition-volume-six/>



HOW-TO

Written by Greg D. Walters

Program In Python - Part 55

This is the second in a multi-part tutorial on creating a Cross Stitch pattern generator. In the first part (FCM85), we created a database containing the DMC™ floss colors with their closest RGB values. In this part, we will create the GUI using Tkinter. We will also use PIL (Python Imaging Library) and PMW (Python Mega Widgets). You'll need to download those libraries and install them before we go too far. For PIL, go to the Pillow fork at <https://github.com/python-imaging/Pillow> and download the latest version. For PMW, go to <http://pmw.sourceforge.net/> and download from there.

You will also need two image files. One is a simple grey rectangle 500x400 pixels. You can use GIMP or some other image manipulating program to create it. Name it default.jpg, and place it into your source code directory along with the database. The other is an image of a folder for the open image button. I got one from open clipart and searched for the word "folder". I found a reasonable one at

<https://openclipart.org/detail/177890/file-folder-by-thebyteman-177890>. Open it in GIMP, resize it to 30x30 and save it in the same directory as the other two files as "open.gif".

Above is a screenshot of what the finished GUI will look like. There are four main frames in the GUI. Three on the left side and one on the right. When we go through the build widget process, I refer to them as Top Frame, Middle Frame, Bottom Frame and Side Frame. The top frame deals with the original image. The middle frame deals with

the processing of the image. The bottom frame shows the original image on the left and the processed image on the right, and the side frame displays the colors and floss required. It seems from first glance there is a lot of wasted space here, but when you see the program run, it doesn't really have that much empty space, once we get through the processing portion.

Now we are ready to start working on the code. Here is our long list of imports...

```
from Tkinter import *
import tkFileDialog
import tkCommonDialog
import tkMessageBox
import ttk

from PIL import
Image, ImageTk, ImageOps

import Pmw

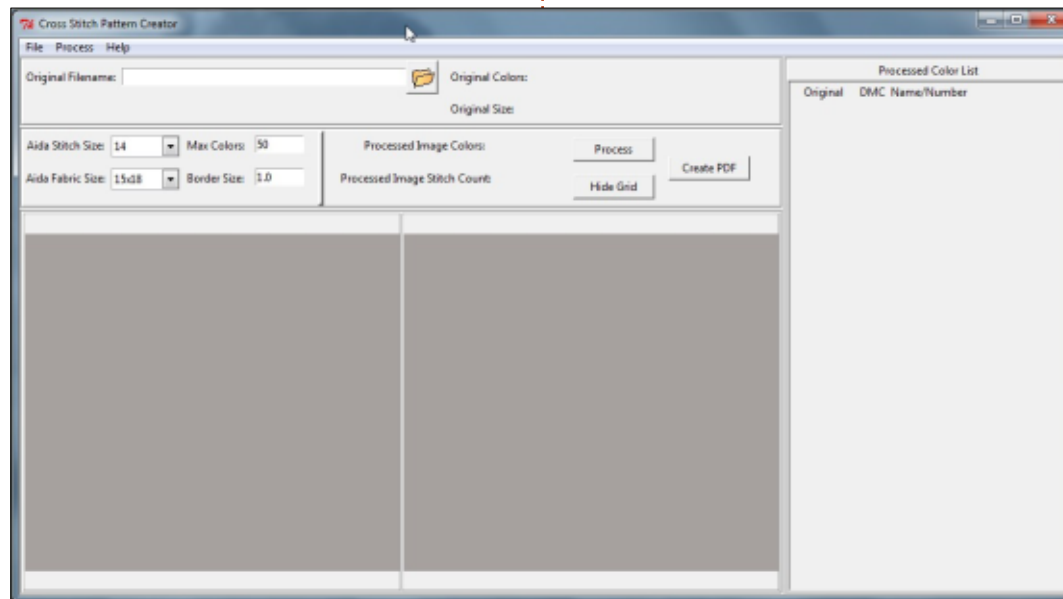
import apsw # Database
Access

import math # Math library

import sys
```

From the sheer number of imports, you can tell this is going to be a long program. In fact, the UI portion of the code will be over 300 lines, including comments. The "good" news is that about 200 of the lines of code deal with the Tkinter portion of the program, the actual GUI itself. The majority of the remaining lines of code in this portion are stubs for functions needed for the next article.

We'll create a class to hold all of our UI processing code (next page,



top right).

First, we have the class definition and next we have the `__init__` function which we pass the TopLevel “root” window into. We create the TopLevel window in the last four lines of the program. Within the `__init__` function we are defining all the global variables and doing some initial assignments before we start the other functions. The first thing we do is create a list of Tuples that hold the picture file formats that we need when we call the OpenFileDialog. The next two lines below, define and ready the two image files we just created (open folder GIF file, and the grey rectangle – which will be used as placeholders for our images used to create the pattern.

```
self.openimage =
PhotoImage(file='open.gif')
```

```
self.DefaultImage
=ImageTk.PhotoImage(self.Thumb
nail("default.jpg",450,450))
```

Now we get into the global definitions (middle right). If you remember, when you use Tkinter, if you have a widget like a text entry box or combo box that you want to retrieve the information selected or entered, you define a global

variable and then assign it to a Variable Class (BooleanVar, DoubleVar, IntVar or StringVar). This will then “track” changes to the values within the widget values so you can access them with the `.get()` or `.set()` methods. In the next lines of code, we create the global variable name, then assign it to the proper wrapper class. I put some comments into the code to try to help you keep track of what we are doing.

As you can see, we are setting a variable called OriginalFilename, which holds the image that we want to create the pattern from, OriginalColorCount which holds the number of colors in the original image file, and OriginalSize which holds the size in pixels of the original image. As they say on tv... ***“BUT WAIT! THERE'S MORE!”*** (bottom right):

The ComboStitch variable is set by a combobox, and handles the stitch size of the aida that you wish to use for your project. The ComboSize variable is also set by a combo box and holds the size of the aida fabric. FabricHeight and FabricWidth are the breakdowns from the aida size. MaxColors is a value from an entry box to set the

```
class XStitch:
    def __init__(self, master):
        self.picFormats = [
            ('JPEG / JFIF', '*.jpg'),
            ('Portable Network Graphics', '*.png'),
            ('CompuServer GIF', '*.gif'),
            ('Windows Bitmap', '*.bmp'),
            ('All File Types *.*', '*.*'),
        ]
```

```
#-----
#                               Global Definitions
#-----
#                               UI Required
global OriginalFilename
OriginalFilename = StringVar()
global OriginalColorCount
OriginalColorCount = StringVar()
global OriginalSize
OriginalSize = StringVar()
```

```
global ComboStitch
ComboStitch = IntVar()
global ComboSize
ComboSize = StringVar()
global FabricWidth
FabricWidth = DoubleVar()
global FabricHeight
FabricHeight = DoubleVar()
global MaxColors
MaxColors = IntVar()
global BorderSize
BorderSize = DoubleVar()
```

number of colors, and BorderSize is a floating point value that specifies the amount of unused aida for framing.

```
global ProcessedColors
ProcessedColors = StringVar()
```

```
global ProcessedSize
ProcessedSize = StringVar()
global DmcColor
DmcColor = StringVar()
```

The final 'variable class' variables are used for information once we have processed the original image to the desired parameters.

The next set of globals is (top right) used for easy access throughout the program. For the most part, they are either obvious by their name, or will become obvious once we use them. There are three not-so-obvious variables here. `backgroundColor1` and `backgroundColor2` are tuples that are used in the gridding process, and the `ReadyToProcess` variable is used to designate that the original image is loaded and everything is ready to go – just in case the user presses the Process button too early.

Finally we have assigned all our globals, and now have the code that actually creates the GUI. We open the database, create the menu, set up the widgets, and finally place the widgets into the proper places. Just to give you a heads-up, we will be using the Grid geometry placement manager. More on that later.

```
#-----
self.OpenDB()

self.MakeMenu(master)

frm =
self.BuildWidgets(master)

self.PlaceWidgets(frm)
```

The next portion of our code (middle right) will set up the menu bar. I've tried to lay it out logically so it will be easy to understand.

We define a function called `MakeMenu`, and pass in the `TopLevel` window. We then define the three menu sets we will be creating. One for File, one for Process, and one for Help.

```
menu.add_cascade(label="File", menu=filemenu)

menu.add_cascade(label="Process", menu=process)

menu.add_cascade(label="Help", menu=help)
```

Now we set up the File menu options (bottom right). Open will open our image and uses a function called "GetFileName". Save will create the output PDF file and uses the `FileSave` function. We add a separator and finally an Exit function.

```
#-----
global ShowGrid
ShowGrid = True
global ProcessedImage
ProcessedImage = ""
global GridImage
GridImage = ""
global backgroundColor1
backgroundColor1 = (120,)*3
global backgroundColor2
backgroundColor2 = (0,)*3
global ReadyToProcess
ReadyToProcess = False
```

```
=====
#
# BEGIN UI DEFINITION
#
=====
def MakeMenu(self, master):
    menu = Menu(master)
    root.config(menu=menu)
    filemenu = Menu(menu, tearoff=0)
    process = Menu(menu, tearoff=0)
    help = Menu(menu, tearoff=0)
```

```
#-----
#
# File Menu
#-----
filemenu.add_command(label="New")
filemenu.add_command(label="Open", command=self.GetFileName)
filemenu.add_command(label="Save", command=self.FileSave)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=self.DoExit)
```

Now we have the Process option and the Help functions (next page, top right).

All of the options in the menu bar are also available from various buttons within the program.

Now we will make our `BuildWidgets` function. This is where we create all the widgets that will be used on the GUI.

```
def BuildWidgets(self, master):
```



```
self.frame =
Frame(master,width=900,height
=850)
```

We start with the function (bottom right) definition, passing in the TopLevel window (master) and placing a frame that holds all of our other widgets. I've added comments to help realize which part of code deals with which frame. We'll deal with the top frame first.

Assuming you remember or refreshed your memory on Tkinter, it should be fairly straight-forward. Let's look at the first label as a discussion item.

```
self.label1 =
Label(self.frm1,text =
"Original Filename: ")
```

First, we define the name of the widget (self.label1 =). Next we set that variable to which widget type we want to use; in this case Label. Finally we set the parameters we want to apply to that widget starting with the parent widget (self.frm1), and in this case, the text that will show up in the label. Now let's take a moment to look at the button self.btnGetFN.

```
self.btnGetFN =
Button(self.frm1, width=28,
```

```
image=self.openimage,
command=self.GetFileName)
```

First thing to notice is that this is broken into two lines. You can safely place everything on one line...it is just too long to fit into a 72-character line. We'll really pay attention to the parameters we use here. First the parent (frm1), next the width which is set at 28. When we use a widget that has the option of text or an image, we have to be careful setting the width. If it will contain text, the width parameter is the number of characters it will hold. If it is to display an image, it will be set at the number of pixels. Finally, we set the command

```
#-----
#                               Process Menu
#-----
process.add_command(label="All",command=self.Process)
#-----
#                               Help Menu
#-----
help.add_command(label="Help",command=self.ShowHelp)
help.add_separator()
help.add_command(label="About",command=self.ShowAbout)
```

parameter, which tells the system what function to call when the button is clicked.

One more thing to look at is the textvariable parameter. This tells us what variable will hold the

information that will be displayed in the widget. We set these in the __init__ function earlier. One other thing to mention is that the frame itself has two parameters you might not remember. The Relief parameter sets the border type of

```
# -----Middle Frame -----
self.frm2 = Frame(self.frame,width=900,height=160,bd=4,relief=GROOVE)
self.lbl4 = Label(self.frm2,text="Aida Stitch Size: ")
self.lbl5 = Label(self.frm2,text="Aida Fabric Size: ")
self.TCombobox1 = ttk.Combobox(self.frm2,textvariable=ComboStitch,width=8)
self.TCombobox1.bind('<<ComboboxSelected>>', self.StitchSizeSelect)
self.TCombobox1['values'] = (7,10,11,12,14,16,18,22)
self.TCombobox2 = ttk.Combobox(self.frm2,textvariable=ComboSize,width = 8)
self.TCombobox2.bind('<<ComboboxSelected>>',self.AidaSizeSelect)
self.TCombobox2['values'] = ("12x18","15x18","30")
```

```
# ----- TOP FRAME -----
self.frm1 = Frame(self.frame,width=900,height=100,bd=4,relief=GROOVE)
self.label1 = Label(self.frm1,text = "Original Filename: ")
self.entFileName = Entry(self.frm1,width=50,textvariable=OriginalFilename)
self.btnGetFN = Button(self.frm1, width=28, image=self.openimage,
command=self.GetFileName)
self.label2 = Label(self.frm1,text = "Original Colors: ")
self.lblOriginalColorCount = Label(self.frm1,text="",width=10,
textvariable=OriginalColorCount)
self.label3 = Label(self.frm1,text = "Original Size: ")
self.lblOriginalSize = Label(self.frm1,text="",width=10,
textvariable=OriginalSize)
```

the frame, which in this case is GROOVE, and the bd parameter sets the border width. Border width defaults at 0 so if you want to see the effect, you have to set the border width (bd is a shortcut).

Now we'll deal with the middle frame widgets.

The last 6 lines of this section (previous page, middle right) deal with the two combo boxes in the UI. Each combo box uses three lines (the way I programmed it to make it easy to understand). In the first line, we set the basic parameters. The next line, we bind the combobox selection-changed event to the function StitchSizeSelect, and the last line has a list of the values that will be available for the pulldown.

Everything else above is pretty "normal" stuff. Now we set our defaults for the widgets that need them. Again, we are using the global variables that we set up in the __init__ function and wrapped to the widget variable class.

`ComboStitch.set (14)`

`ComboSize.set ("15x18")`

`FabricWidth.set (15)`

```
self.lbl6 = Label(self.frm2,text="Max Colors: ")
self.entMaxColors = Entry(self.frm2,textvariable=MaxColors,width=3)
self.lbl7 = Label(self.frm2,text="Border Size: ")
self.entBorderSize = Entry(self.frm2,textvariable=BorderSize,width = 8)
self.frmLine = Frame(self.frm2,width=6,height=80,bd=3,relief="raised")
    self.lbl8 = Label(self.frm2,text="                Processed Image Colors: ")
self.lbl9 = Label(self.frm2,text="Processed Image Stitch Count: ")
self.lblProcessedColors = Label(self.frm2, width=10,textvariable=ProcessedColors,
    justify=LEFT)
self.lblProcessedSize = Label(self.frm2, width=10, textvariable=ProcessedSize,
    justify=LEFT)
self.btnDoIt = Button(self.frm2,text="Process",width=11,command = self.Process)
self.btnShowGrid = Button(self.frm2,text="Hide Grid", width=11,
    command=self.ShowHideGrid)
self.btnCreatePDF = Button(self.frm2, text="Create PDF", width=11,
    command=self.CreatePDF)
```

```
# ----- Bottom Frame -----
self.frm3 = Frame(self.frame,width=450,height=450,bd=4,relief=GROOVE)
self.lblImageL = Label(self.frm3, image=self.DefaultImage,
    height=400, width=400, borderwidth=2, relief=GROOVE)
self.lblImageR = Label(self.frm3, image=self.DefaultImage, height=400,
    width=400,borderwidth=2, relief=GROOVE)
```

`FabricHeight.set (18)`

`MaxColors.set (50)`

`BorderSize.set (1.0)`

Now we deal with the bottom frame. This is really simple, since

we have to set up only the frame and two labels which we will use to hold our images.

Finally we deal with the side frame. The side frame will hold a ScrolledFrame from the PMW

library. It's really easy to use and provides a nice interface to the information about the floss that should be used. You can research the ScrolledFrame on your own, since we still have a lot to cover here.

```
#----- Side Frame -----
self.frm4 = Frame(self.frame,width = 300,height=580,bd=4,relief=GROOVE)
# Create the ScrolledFrame.
self.sf = Pmw.ScrolledFrame(self.frm4,
    labelpos = 'n', label_text = 'Processed Color List',
    usehullsize = 1,
    hull_width = 300,
    hull_height = 567,)
return self.frame
```

That's all for the widgets. Now we have to place them. As I said earlier, we will be using the Grid geometry manager, rather than the absolute or pack managers.

The Grid method places the widgets in (you guessed it) a grid, referenced by row and column designations. I'll use the top frame as an example (shown top right).

First we place the frame.

You can see that we place the widget by using the {widgetname}.grid command, then the row and column positions. Notice that we are telling the entry widget to span 5 columns. Padx and pady values will place some extra space on both the right and left sides (padx) or the top and bottom (pady). The sticky parameter is similar to a justify command for text.

The middle frame is a bit more complicated, but basically the same as the top frame. You might notice an extra frame in the middle of the code (self.frmLine). This gives us a nice divider between the options section and the display section. Since there is no horizontal or

ROW	Col 0	Col 1 - Col 6	Col 7	Col 9	Col 10	
0	Label1	entFileName	btnGenFN	Label2	lblOriginalColorCount	
1				Label3	lblOriginalSize	


```

def PlaceWidgets(self, frame):
    frame.grid(column = 0, row = 0)
    # ----- TOP FRAME -----
    self.frm1.grid(column=0, row=0, rowspan=2, sticky="new")
    self.label1.grid(column=0, row=0, sticky='w')
    self.entFileName.grid(column=1, row=0, sticky='w', columnsnpan = 5)
    self.btnGetFN.grid(column=7, row = 0, sticky='w')
    self.label2.grid(column=9, row=0, sticky='w', padx=10)
    self.lblOriginalColorCount.grid(column=10, row=0, sticky='w')
    self.label3.grid(column=9, row=1, sticky='w', padx=10, pady=5)
    self.lblOriginalSize.grid(column=10, row=1, sticky='w')

    # ----- MIDDLE FRAME -----
    self.frm2.grid(column=0, row=2, rowspan=2, sticky="new")
    self.lbl4.grid(column=0, row=0, sticky="new", pady=5)
    self.lbl5.grid(column=0, row=1, sticky="new")
    self.TCombobox1.grid(column=1, row=0, sticky="new", pady=5)
    self.TCombobox2.grid(column=1, row=1, sticky="new")
    self.lbl6.grid(column=2, row = 0, sticky="new", padx=5, pady=5)
    self.entMaxColors.grid(column=3, row=0, sticky="new", pady=5)
    self.lbl7.grid(column=2, row=1, sticky='new', padx=5)
    self.entBorderSize.grid(column=3, row=1, sticky='new')
    self.frmLine.grid(column=4, row=0, rowspan=2, sticky='new', padx=15)
    self.lbl8.grid(column=5, row=0, sticky='new', pady=5)
    self.lbl9.grid(column=5, row=1, sticky='new')
    self.lblProcessedColors.grid(column=6, row=0, sticky='w')
    self.lblProcessedSize.grid(column=6, row=1, sticky='new')
    self.btnDoIt.grid(column=7, row=0, sticky='e', padx=5, pady = 5)
    self.btnShowGrid.grid(column=7, row=1, sticky='e', padx=5, pady = 5)
    self.btnCreatePDF.grid(column=8, row=0, rowspan=2, sticky='ew', padx=10)

    # ----- BOTTOM FRAME -----
    self.frm3.grid(column=0, row=4, sticky="nsew")
    self.lblImageL.grid(column=0, row=0, sticky="w")
    self.lblImageR.grid(column=1, row=0, sticky="e")

```

vertical line widget, I cheated and used a frame with a width of 6 pixels and border width of 3, making it just look like a fat line.

The bottom frame is simple since we have only the frame and the two labels to hold the images.

The side frame is pretty much the same thing, except the ScrolledFrame allows for a frame to be set to the interior of the scrolled frame widget. We then create three widgets here and place them in their grids as column headers. We do this since we assigned the interior frame for the scroll frame here and we have to assign the parent (self.sfFrame) after we have created it.

That's all the hard work for now. At this point, we will create all of the functions that we need to get the GUI to run, stubbing most of them until next month. There are a few we will go ahead and complete, but they are fairly short.

The first function will be the Exit option from the menu bar. It's under the File menu option.

```
def DoExit(self):  
    sys.exit()
```

The only other one is the Thumbnail function. We need this to fill the grey rectangles into the labels in the bottom frame. We

pass the filename and the width and height that we want the thumbnail image to be.

Since this article is so long, I'm going to give you a list of function names and all you have to do is stub it out by using the pass command. We'll fill them in next month. I'll give you the first one as an example, but you should already know how to do it.

```
def GetFileName(self):  
    pass
```

For the rest of the functions, I'll just give you the def lines. Be sure to include them all in your code.

You can see, we have a large amount of work to do next month. We still have four more lines to write to finish up for this month. This is out of our class code.

```
root = Tk()  
root.title("Cross Stitch  
Pattern Creator")
```

```
def ShowHelp(self):, def ShowAbout(self):, def OpenDB(self):, def ShowHideGrid(self):  
def StitchSizeSelect(self,p):, def AidaSizeSelect(self,p):, def Process(self):  
def CreatePDF(self):, def OriginalInfo(self,file):, def GetColorCount(self,file):  
def GetHW(self,file):, def GetHW2(self,file):, def GetColors(self,image):  
def Pixelate(self,im,pixelSize):, def ReduceColours(self,ImageName):  
def MakeLines(self,im,pixelSize):, def MakeLines2(self,im,pixelSize):  
def Rgb2Hex(self,rgb):, def FillScrolledList(self,filename):  
def GetBestDistance(self,r1,g1,b1):
```

```
# ----- SIDE FRAME -----  
self.frm4.grid(column=2,row=0,rowspan=12,sticky="new")  
self.sf.grid(column=0,row=1)  
self.sfFrame = self.sf.interior()  
self.lblch1 = Label(self.sfFrame,text="          Original")  
self.lblch2 = Label(self.sfFrame,text="          DMC")  
self.lblch3 = Label(self.sfFrame,text="Name/Number")  
self.lblch1.grid(column=0,row=0,sticky='w')  
self.lblch2.grid(column=1,row=0,sticky='w')  
self.lblch3.grid(column=2,row=0,sticky='w')
```

```
def Thumbnail(self,file,hsize,wsize):  
    size = hsize,wsize  
    extpos = file.rfind(".")  
    outfile = file[:extpos] + ".thumbnail"  
    im = Image.open(file)  
    im.thumbnail(size)  
    im.save(outfile,"JPEG")  
    return im
```

```
test = XStitch(root)  
root.mainloop()
```

The first line sets up the root TopLevel window. The next line sets the title on the top line. The third line instantiates our XStitch class, and the last line starts the main loop that shows the UI and gives control over to it.

Well that's a lot for this month, but we are finally done. You can actually run the program to see the GUI.

As always, the code is available on Pastebin at <http://pastebin.com/XtBawJps>.

Next month we will flesh out the code. See you then.



HOW-TO

Written by Cork Marino

LibreOffice Pt.38 - Contents and Indexes

Imagine a scientist, let's call him Doc Brown, who has just written a manuscript for a book describing his new theory on time travel. The manuscript is a few hundred pages long. He has broken it down into chapters and sections, but he needed to add a table of contents and an index that the other scientists, who will praise and adore his work, can use to navigate his book with ease. Luckily, he knew about LibreOffice and how to use Writer's indexes and tables tools to create a table of contents and an index. Let's see how he did it.

SETTING UP THE STYLES

In many of my articles, I have emphasized the importance of using styles, but just in case you haven't got the message yet, "Using styles is the best way to save time and create uniformity in your documents." Luckily, Doc Brown knew the importance of using styles, and he used styles to help simplify the creation of his table of contents. These are the heading styles "Heading 1[...10]."

He could edit the styles in any way he wanted for appearance, but we are interested in how he used them in the overall outline of the document.

To set up the overall outline of his document, Doc Brown opened the "Outline Numbering" dialog, Tools > Outline Numbering. Selecting each of the different levels, he noticed that each one was already assigned to a heading style in order from 1 to 10.

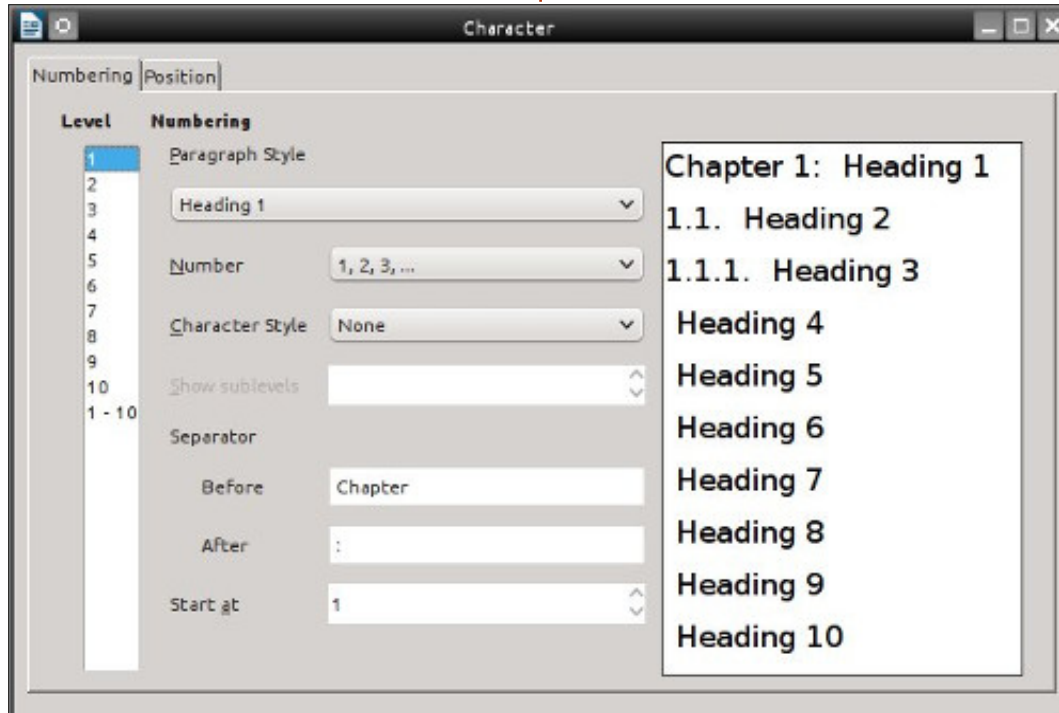
Working with these defaults he selected level 1 with the paragraph style "Heading 1" assigned as its style. For the "Number" setting, he selected "1, 2, 3, ..." For the "Before", he entered "Chapter " (note the space), and for the "After", he entered a colon ":". Using this method, each "Heading 1" inserted into the document would have the format of "Chapter N: Title", where N is the chapter number.

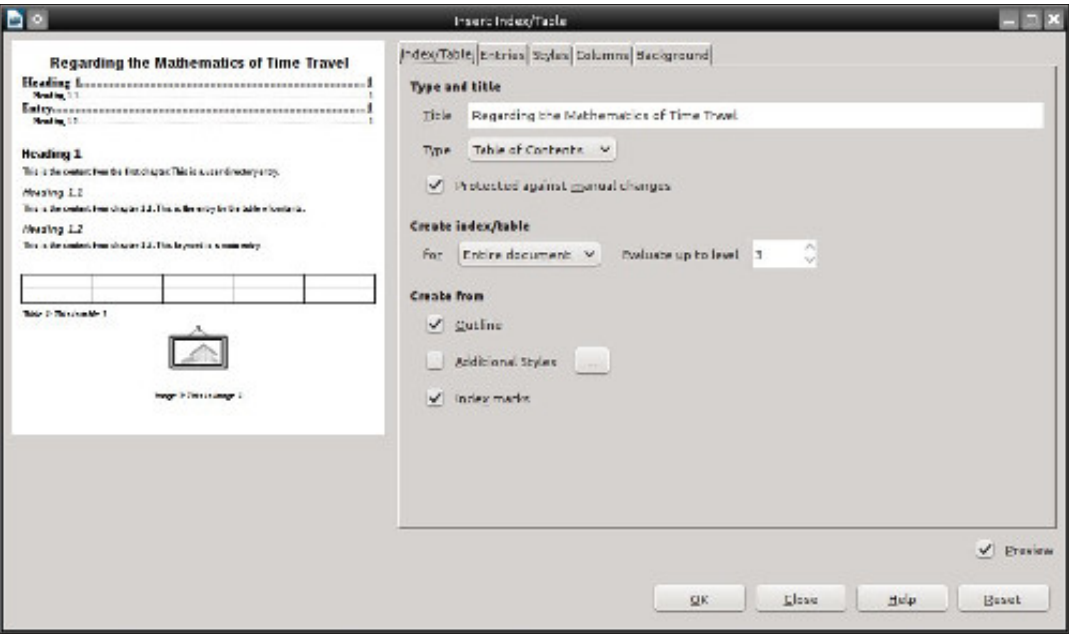
Doc Brown has also decided that the second and third levels needed a numbering scheme of 1.1 for the second level and 1.1.1 for the third level. On the second level, he changed "Number" to "1, 2, 3, ...", set "Show sublevels" to 2, and added a period and space to "After." For the third level, he set "Number" to "1, 2, 3, ...", set "Show sublevels" to 3, and added a period and space to "After."

With all the styles set, Doc Brown went to work on his document. He used "Heading 1" for all the chapter headings, and "Heading 2" and "Heading 3" for the subheadings within the chapters. Because he took the time to set up his styles, he could easily create his table of contents once his document was finished.

CREATE THE TABLE OF CONTENTS

With the writing all done, Doc Brown was ready to create his table of contents. Placing the cursor below the last line of the





title page, he decided the table of contents should start on a new page. Insert > Manual Break brought up the “Insert Break” dialog. He selected “Page Break” and clicked OK. The cursor moved to the start of a new page.

Now, to create the table of contents. Insert > Indexes and Tables > Indexes and Tables. The “Insert Index/Table” dialog appeared. He changed “Title” to “Regarding the Mathematics of Time Travel.” For the “Type”, he made sure that “Table of Contents” was selected.

The “Create Table/Index for”

gave him two options: “Entire Document” or “Chapter.” If he wanted, he could have placed a table of contents at the beginning of each chapter by inserting a table of contents after the chapter headings and selecting “Chapter” for the “Create Table/Index for.” However, he was creating the table of contents for the whole book, so he selected “Entire Document.”

Finally, he had the “Evaluate up to level” option. Here he selected the depth of his table of contents. If he changed this to 1, only the chapter titles would show. He could then create a table of contents for each chapter to show

the subheadings. In the end, he decided to show 3 heading levels in the table of contents and set this option to 3.

Doc Brown clicked OK, and the table of contents was created. Without any editing, the page numbers for the first three levels were added to the table of contents. Each new level was indented just a little to show that it was a lower level. Each title in the table of contents was a link to the heading in the document. This became handy when he converted the document to PDF and Ebook. He could have left his table of contents like this and it would have worked great, but he decided it needs just a few more added touches.

EDIT THE TABLE OF CONTENTS STYLES

To change the appearance of the different levels in the table of contents itself, Doc Brown would need to edit the “Contents 1[...10]” styles. There was also the “Contents Heading” style which was used to style the table of contents title.

Doc Brown wanted the title centered, enlarged, and bold. He opened the “Styles and Formatting” dialog through the new sidebar (no longer experimental in version 4.2). He could also open the “Styles and Formatting” dialog by clicking on its button in the “Formatting” toolbar, or through the menus, Format > Styles and Formatting. He right-clicked the “Content Heading” style and selected “Modify”. On the “Font” tab, he selected bold and set the size to 20pt. He switched to the “Align” tab and selected “Center.” He clicked OK to save his changes.

Regarding the Mathematics of Time Travel	
Chapter 1: Speed, The Foundation of Time Travel.....	3
1.1. Basic Concepts.....	3
1.1.1. Correlation Between Time and Speed.....	3
1.1.2. The Gain / Speed Ratio of the Flux Capacitor.....	3
1.1.3. The 88 MPH Sweet Spot.....	3
1.1.4. Trial Delorean Test.....	4

Since Doc Brown was only including the first three levels, he only needed to change the paragraph styles "Contents 1", "Contents 2", and "Contents 3." The first level he wanted a little bigger and bold, since these were the chapter titles. He modified the "Contents 1" style by selecting bold and changing the size to 16pt on the "Font" tab. To distinguish the second level from the third, he set the second to bold but left the size at the default. For the third, he changed the font to italic but left the size at the default.

Doc Brown saved his document with its newly formatted table of contents. He was now one step closer to publishing his book and becoming world famous. He smiled as he imagined the raving peer reviews it would receive, but wait... what about the alphabetical index?

CREATE THE INDEX ENTRIES

In a book of such scientific importance, an index at the end would help researchers find a reference to certain scientific knowledge contained in the book. With the final edit completed, Doc Brown began reading through his

manuscript to determine which words he would need to include in the index.

Before he created the index itself, he needed to create the index entries for the words he decided to include in the index. To get started, he selected the first word he wanted indexed; then from the menus, he selected Insert > Indexes and Tables > Entry. The "Insert Index Entry" dialog appeared.

He left the "Index" at the default, "Alphabetical Index." If he had planned to create a custom index, he would create one using the button beside the dropdown list.

"Entry" is the word or phrase for the index, and it doesn't



necessarily have to appear the same as in the manuscript. For example, he changed the term "supplementary angle" to "angle, supplementary" in the index. Even though the order of the words has changed, it continues to refer to the same place in the manuscript.

Writer allows for two levels of "Keys" or categories for grouping indexes together. For example, Doc Brown decided that all indexes concerning angles should appear together in the index, as well as the different operators used. For the angle entries, he entered "angle" in "Key 1", and for the operator entries, he entered "operators" in "Key 1." Though he only created one key level in each case, he could have created two by entering a second value in "Key 2".

Whenever he was on the page where the main text for a topic appeared, he would check "Main Entry." This makes the page number appear in bold numerals for that entry by default. (You can change the appearance of the main entry page number by editing the character style "Main index entry.")

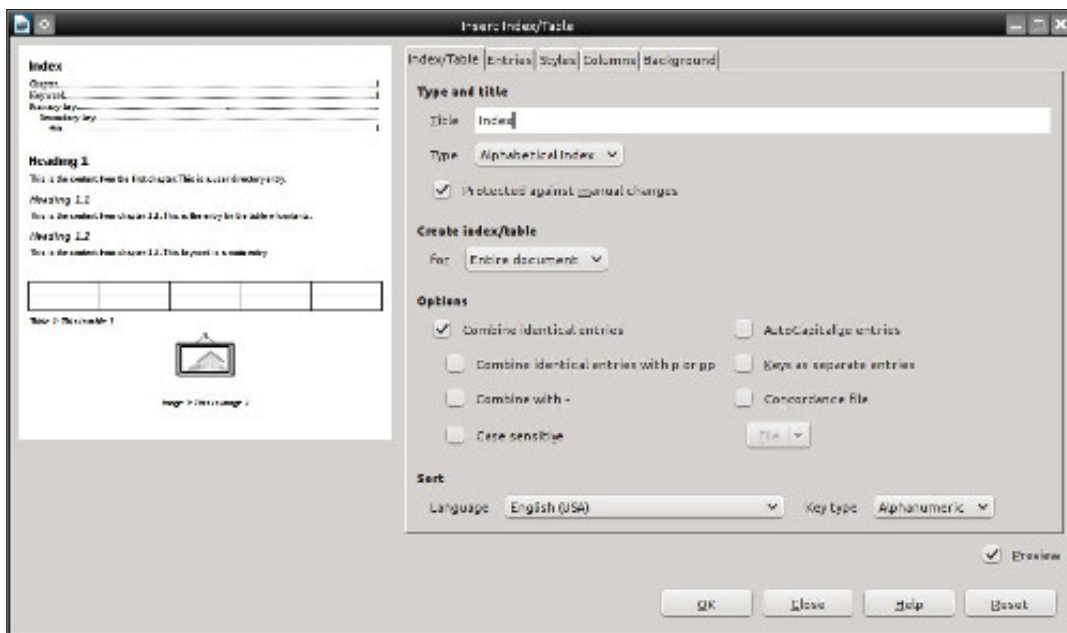
Doc Brown checked the entry "Apply to all similar text" as this

creates an entry for every time the word or phrase appears in the manuscript. He left "match case" unchecked, as sometimes the word or phrase appears in a different case. He checked "Whole words only" as he does not want variations to the word or phrase indexed.

Once finished with a word, Doc Brown clicked "Insert." Leaving the dialog open, he selected the next word or phrase in his document. When he clicked on the dialog, the new word or phrase appeared in the dialog's "Entry" field. He made the changes needed for this entry and clicked "Insert." This ability to switch between the dialog and the manuscript makes creating the index entries quicker than if he had to open the dialog each time.

CREATING THE INDEX

Once Doc Brown completed creating his index entries, he was ready to create the actual index. He went to the last page of his manuscript, and deciding the index should start on a new page, inserted a page break (Insert > Manual Break; select "Page Break"; click OK).



Insert > Indexes and Tables > Indexes and Tables, brought up the "Insert Index/Table" dialog. On the "Index/Table" tab, he selected "Alphabetical Index" for the "Type", changed the "Title" to "Index", and unchecked all entries except "Combine identical entries." This prevents the index from creating separate entries because

of upper or lower case.

Doc Brown decided that the index should have two columns to conserve some space. On the "Columns" tab, he changed the number of columns to 2. To create some space between the two columns, he changed the spacing to 0.20" (0.50cm).

Index	
angles.....	BEDMAS..... 9, 16
Alternate interior angles..... 11	operators.....
Alternating external angles..... 12	addition..... 3, 4, 9, 15, 16
angles, right..... 12, 13	division..... 4, 5, 6, 9
angles, supplementary..... 10, 11, 12	Multiplication..... 3, 4, 5, 6, 9
Complimentary angles..... 10	subtraction..... 3, 9, 15, 16
Corresponding angles..... 10, 11	proportions..... 6, 7, 8, 9
vertically opposite angles..... 11	Speed..... 3, 8

Satisfied the index was set up the way he wanted it, he clicked on OK. The index then appeared on the page the way he formatted it.

Now, Doc Brown's manuscript was ready for publication. He began to dream of the raving peer reviews he would receive. Thanks to LibreOffice, he was able to add his table of contents and index rather quickly.



The Ubuntu Podcast covers all the latest news and issues facing Ubuntu Linux users and Free Software fans in general. The show appeals to the newest user and the oldest coder. Our discussions cover the development of Ubuntu but aren't overly technical. We are lucky enough to have some great guests on the show, telling us first hand about the latest exciting developments they are working on, in a way that we can all understand! We also talk about the Ubuntu community and what it gets up to.

The show is presented by members of the UK's Ubuntu Linux community. Because it is covered by the Ubuntu Code of Conduct it is suitable for all.

The show is broadcast live every fortnight on a Tuesday evening (British time) and is available for download the following day.

podcast.ubuntu-uk.org



Elmer Perry's history of working, and programming, computers involves an Apple][E, adding some Amiga, a generous helping of DOS and Windows, a dash of Unix, and blend well with Linux and Ubuntu. He blogs at <http://eeperry.wordpress.com>



HOW-TO

Written by Floris Vanderhaeghe

GRUB2 & Multibooting Pt.2

In the first part of this mini-series, I explained the basic functionality of GRUB (version 2) and its configuration file `grub.cfg`. Ultimately, we want to achieve a multibooting system where changes or new installs of operating systems (OSs) don't mess up GRUB. Your GRUB menu should offer a choice between multiple installed OSs to choose from. A prerequisite is an essential knowledge of how GRUB works. Follow me to find out!

For now, let's consider a normal installation of GRUB on the Master Boot Record (MBR) of your hard drive, which, for example, has been arranged for you when you installed Ubuntu. If you have a spare machine on which you can play with this GRUB installation, then do it! You can learn a lot from experimenting with GRUB (see also the links at the end of this article). If not, just believe the explanations below, or be more careful in your experiments – the exact nature of your experiments will determine not only how GRUB works, but also whether it still works...

We already learned that GRUB runs from the MBR at hard disk boot, and that it relies on `grub.cfg` which is in `/boot/grub/` on a partition that was specified to GRUB during its installation. Well actually, this "dependence" on `grub.cfg` is needed only for displaying the user-friendly menu from which you can select an OS using arrow keys and Enter. GRUB itself can thrive without `grub.cfg`, although it still needs to load specific modules from `/boot/grub/` in order to access all of its functionality. You can access 'bare' GRUB from the GRUB menu by pressing 'c' (press Esc to return to the menu). Welcome to the GRUB command prompt! The GRUB operating system is just awaiting your commands now – and that puts you in control! This will be useful when, for any reason, the GRUB menu doesn't load, or a GRUB menu entry does not work. An in-depth discussion of the possibilities is far beyond the scope of this article, and I refer to the internet links given at the end. I will discuss some introductory commands at the end of the article

though, just to get you in play... Yes, ladies and gentlemen, mastering the command line opens a great new world of possibilities – also in GRUB!

The GRUB menu will be our focus now. Its components are controlled by the contents of `/boot/grub/grub.cfg`. Have a good look at `grub.cfg`. It is the file that GRUB reads. What is in there? Nothing but GRUB commands! The official manual on gnu.org denotes this as "GRUB's built-in scripting language". By looking carefully, you may already understand a lot of what is stated there: several modules are loaded with `insmod`, for example to handle `msdos`-partitions, or to display `png` or `jpeg` images. Further, several so-called 'menu entries' (lines of the GRUB menu) are listed by their name of the OS plus kernel version, each followed by the specific instructions that GRUB needs to boot the corresponding OS. Essential instructions appear below each menu-entry in order to boot an OS: the hard drive and partition numbers where that OS resides,

and the directory path and name of the kernel to be booted and of the initial ramdisk image (which is loaded first). The GRUB command to point at the Linux kernel is... `linux`.

I'm sure at this stage you do not want to edit `grub.cfg`, and it is not recommended either. So have no worries: the GRUB configuration file is created for you when you type '`sudo update-grub`' in a terminal. The `update-grub` command is delivered with the `grub-pc` package, which was also responsible for installing GRUB on the MBR. The `update-grub` command reads the contents of several files: on the one hand `/etc/default/grub`, which contains several personal settings, and on the other hand the executable scripts under `/etc/grub.d/`. We will play with the scripts next time.

For now, let's tweak `/etc/default/grub`. The first part of this file looks like the one shown on the next page, top right.



HOWTO - GRUB & MULTIBOOTING

- the menu entry that is selected by default when the GRUB menu shows, is determined by `GRUB_DEFAULT=x`. You have to fill in the number of the line, counting from... zero. Bummer.

- the counter (in seconds) after which the default entry is executed if no key is pressed, is determined by `GRUB_TIMEOUT` (as seconds).

- `GRUB_HIDDEN_TIMEOUT=0` should be commented out with a hash (#) if you don't want the GRUB menu to be hidden. This may need to be changed for your first installed Linux distribution if you did not install Windows.

- `GRUB_CMDLINE_LINUX_DEFAULT` sets the kernel parameters to be added to the linux command, for each default menu entry for Linux distros in `grub.cfg`. Recovery entries excluded.

- adding `GRUB_BACKGROUND=/path/filename` determines the splash image to be displayed as a menu background. Alternatively, just putting a background image (jpeg or png) into `/boot/grub` should suffice to be picked up by `update-grub`.

Change these settings as you like. Then comes the moment to update `/boot/grub/grub.cfg` correspondingly. First, backup your existing `grub.cfg`. Then run this in a terminal:

```
sudo update-grub
```

You will see some messages during the process, which should last less than a minute. After that, have a look at `grub.cfg` to check if everything looks OK. Reboot and enjoy your changes!

Booting with an erratic (or absent) `grub.cfg`, will leave you in the GRUB terminal, or – worse – in GRUB rescue mode. So it is a very good idea to be prepared for that (see the online resources). The best way to prepare for this is to run exercises in the GRUB terminal and learn to boot OSs from there. You better do that before experimenting with `grub.cfg` (so that you still have a working GRUB menu at your disposition). When you see the GRUB menu during booting, press 'c' to get at the GRUB prompt (and Esc to turn back to the menu). Now you can learn to

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'
GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=4
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX=""
```

manually run GRUB commands (instead of letting the menu do this for you). I list only a few commands here to start with:

```
set pager=1 # to avoid longer
GRUB output to roll off your
screen
```

```
help [command] # without
argument, lists available
commands; with argument,
shows help of the specified
command
```

```
set # lists current settings
```

```
ls <path> # displays contents
of a directory
```

```
cat <file> # displays
contents of a file
```

```
boot # starts the boot
sequence; identical to
selecting an OS in the GRUB
menu with Enter.
```

This command will work only if GRUB has been told where the OS to be booted resides: you find

these essential commands under each 'menu entry' in `grub.cfg`, or just wait till next time!).



Floris Vanderhaeghe became a Linux fan through Ubuntu 10.10. You can email him at tux7546@gmail.com.





HOW-TO

Written by Nicholas Kopakakis

Blender Part 11c

Ok. Now we have to set up our scene.

For this project I prefer the Blender renderer so if you are on Cycles change it to blender.



First of all we have to set up the world. So navigate to the world's properties panel and change the Horizon color to solid black.



Then we need to adjust our camera.

Select the camera and if it's hidden press N to open the properties panel. Set X, Y and Z to 0 and the rotation of X-axis to 75 degrees.

Now, we want to use the image of the star wars logo that we downloaded earlier.

So, add a plane, create a material, and, under texture, select Image or Movie. Then open the starwars logo. At the and you must have something like that for the material and the texture of our plane.

One adjustment that you have to make is the specular color to black under the material tab and under the texture tab you have to change the default Image Mapping Extension from Repeat to Extend.

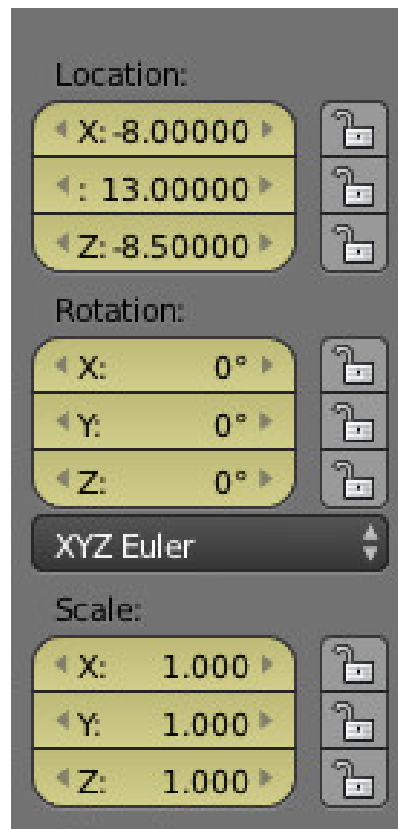
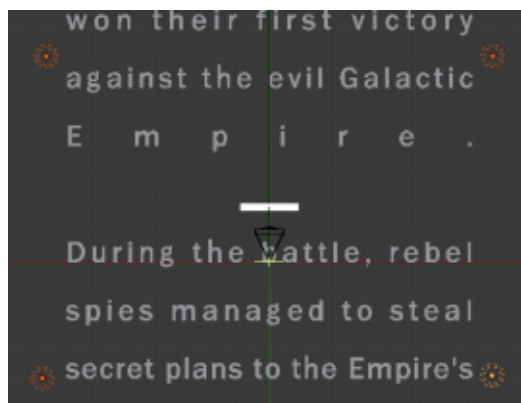
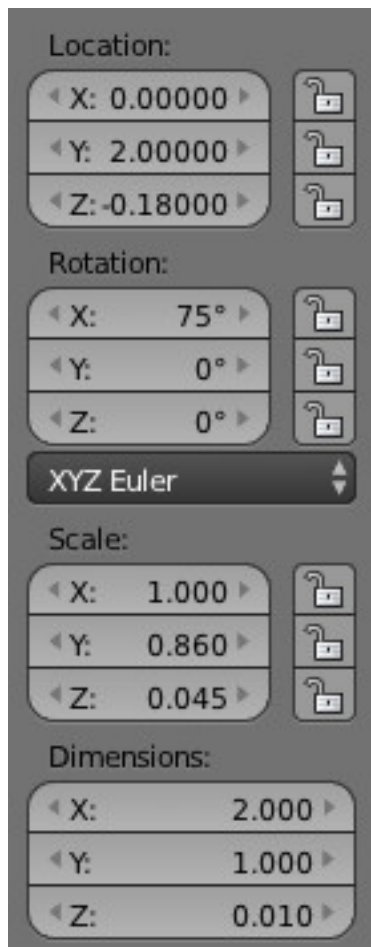
Now we have to position our logo towards camera as the properties panel (next page, left)



Another issue is the lights of the scene. As we want a general luminance I suggest 4 lamps as a square. Something like the image shown (next page, bottom left).

Finally we want to adjust our crawling text object.





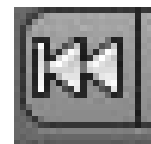
Also, we have to add a material to our crawling text.

Select the crawling text object and under the material tab change the Diffuse color to Red: 0.898, Green: 0.694 and Blue: 0.227. What are these numbers, you asked? Well, we have the colors noted from <http://www.theforce.net/fanfilms/postproduction/crawl/opening.asp> website.

What we get from there? Red: 229, Blue: 177 and Green: 58. To

translate these numbers to color percentage that Blender uses for RGB do the maths in Blender as the image below (Remember that the color range is between 0-255).

So, $229/255$ is 0.898 for Red, $177/255$ is 0.698 for Green and $58/255$ is 0.227 for Blue.



Now that every object is in place it's time for our animation. Press the leftmost button of the timeline navigation to ensure that you are in frame 1

Now press A to select everything in your scene and press the I key to insert a key frame. From the list select LocRotScale.

Go to frame 200 or enter 8 (seconds) X 25 (frames)



Select the Star Wars logo object (the plane that we created earlier) and change its position properties to 100 for Y axis and -26 for the Z axis.

Press the I key again to insert a new key frame.

Now, select the crawling text object. Go to last frame by pressing the rightmost button of the timeline navigation buttons and change its position properties to 142 for the Y axis and to -24 for the Z axis.

Press the I key once more to insert a new key frame to the last frame.

That's it. We are done with our extra super-simple animation. But if you can understand that animation is all about key frames, you can animate anything.

Save your project.

We are ready to render. But this is something that we will talk about next month.



Nicholas lives and works in Greece. He has worked for a post-production house for several years and migrated to Ubuntu because "it renders faster." You can email him at: blender5d@gmail.com



HOW-TO

Written by Mark Crutch

Inkscape - Part 24

Inkscape's Spray tool is the vector graphics equivalent of similar tools from the bitmap world. It's used to create semi-random arrangements of objects but unlike the bitmap version, each object can then be manipulated individually like any other vector element. This makes it particularly good for quickly filling large areas with similar items – think of a snow flurry, or a path covered in autumn leaves – but it can also be invaluable on a smaller scale for textured outlines and shapes.



The Spray tool is activated by clicking on the toolbox icon or by pressing either the “A” key or SHIFT-F3. As usual, the tool control bar lets you modify the behaviour of the tool using buttons and sliders, the latter with right-click context menus that expose a sensible range of values and, perhaps more importantly, label the default values so you can quickly get back to something sensible.

To use the spray tool you first



need an object to spray. I've used a simple leaf design for this example, made up of some paths grouped together. Next, you need to select your object using the select tool before switching to the Spray tool. With the first “Mode” button selected, and all the sliders at their default values, move the cursor into the canvas area, press and hold the left mouse button, and move the mouse around. Copies of your object should be sprayed onto the screen with random scale and rotation, which in my case produces a pile of leaves (with the original leaf on the left).



The important thing to note is

that each leaf created by the Spray tool is an independent object that can be further manipulated. Don't like the position of one of the leaves? Move or delete it. If the size is wrong, scale it using the Select tool. Rotate it, change the fill and stroke colors, move it up or down in the Z-index, or group it with a few neighbours. All the Spray tool has done is the same job you could have completed by copying the original object then pasting it multiple times, with some simple adjustments to each one.

If the Spray tool is just a fancy way of doing a quick copy and paste job, what happens when you have more than one object selected? In this example I've manually made two copies of the leaf design and changed the colors. Then I've selected all three and switched to the Spray tool, using the same parameters as before.

You could probably have



guessed that I would get all three leaves sprayed onto the canvas, but look at their relative positions and rotations: each of them is sprayed independently of the others. The original trilogy forms a line with all the stalks pointing in the same direction, but the sprayed result doesn't preserve that relationship between the items. In this case that's exactly what I want – my leafy background wouldn't look quite the same if it was made up entirely of regimented triplets – but if you do want that effect you have only to group your original objects first. That way the Spray tool is dealing with only a single

HOWTO - INKSCAPE

(compound) object, rather than three separate ones.

Now that you've got the basics of the tool, let's look at the controls that are available and the effect that each of them has. Starting at the left, the Mode buttons are arguably the most important as they dictate the way in which your final result is actually structured – whether you'll end up with real objects, clones, or a single complex path.

One of those words, “Clones”, is a new one in this series, and it's a subject that I'll be covering in a lot more depth in future articles. For now it's sufficient to know that a clone is like a duplicate of an object that retains a live link to the original: any changes made to the original are immediately reflected in the clones. Consider trying to change the colors of the leaves in our pile. With each sprayed leaf as a copy of the original you would need to dissect the pile and re-color each leaf individually. If, however, you select the second Mode button in order to create clones instead, then changing the color requires you to modify only the original. Any changes to the original are propagated, so you can

even enter the group and tweak the paths to change the leaf shape, with all your modifications immediately reflected in the sprayed versions.

When to create clones and when to create copies is sometimes hard to judge. As a general rule of thumb, though, it's usually safer to create clones as they can subsequently be converted into real copies by selecting them and then using the Edit > Clone > Unlink Clone menu entry, whereas you can't convert in the opposite direction. In that same menu, the Select Original item will select the original “parent” object for the currently active clone – a trick that

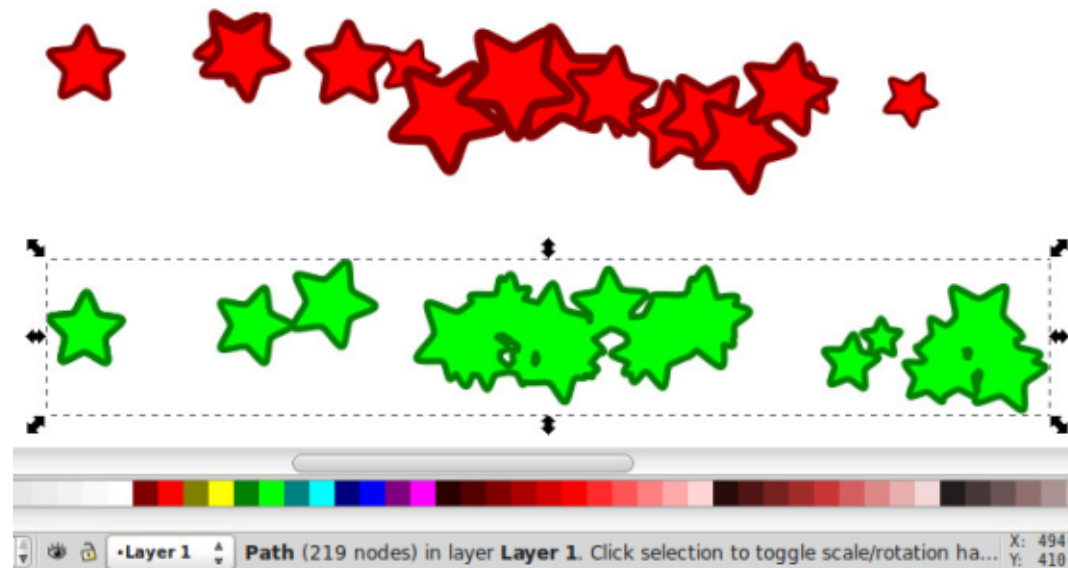
can be invaluable when your pile of leaves grows large enough to obscure the originals.

The third Mode button works only when the object you're spraying is a single path. Instead of creating separate objects or clones, it adds each sprayed item as part of a single complex path. In this image, the red stars have been created as clones of the original, whereas the green stars have been created using this “Path” mode. As you can see, the areas where sprayed objects overlap differs considerably, and in Path mode the end result is a single path which includes even those stars that appear to be separate.

Having discussed the Mode buttons, it's time to move on to the remaining controls on the tool control bar. With the exception of one misplaced button, these are all sliders which run up to 100 and down to 1 or 0, depending on the control.

The first slider, Width, simply alters the size of the spray area. This value is reflected in the size of the orange circle that surrounds the cursor when it's in the canvas area. You can think of this circle as containing all the possible locations that could be used to place the center of the sprayed shape. Keeping it small lets you constrain the spray close to the cursor, whereas a large value places the objects over a wider area that is just centered on the cursor.

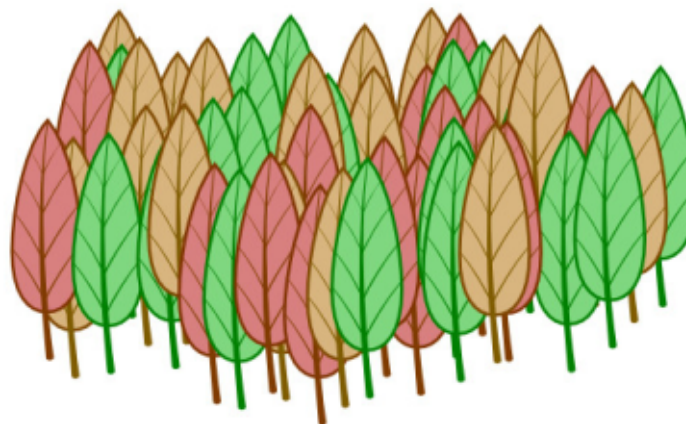
The next slider, Amount, is used to adjust the “speed” of your spray can, or the number of objects that are created over a particular period. The button to the left can be used to set whether or not the Amount value is affected by the pressure of the stylus on a pressure-sensitive graphics tablet. You may recall similar buttons from



the Calligraphy tool (see Part 18 of this series) and Tweak tool (Part 23), but in both those cases the button is positioned to the right of the slider it controls, rather than the left. This inconsistency in UI is likely just an oversight, but if you do use a graphics tablet, it's worth checking the tooltips for these buttons to confirm exactly which controls they affect.

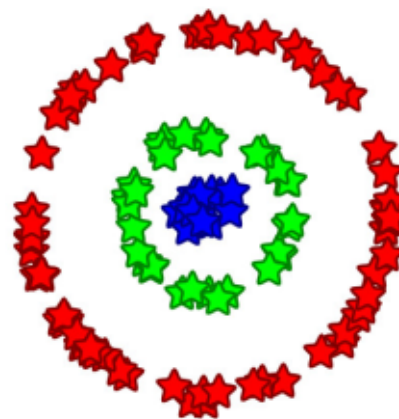
The Rotation and Scale controls are pretty obvious. Just be aware that the values of the sliders use an arbitrary scale running from 0 to 100, rather than just showing the real numbers they represent: for Rotation, the slider sets the maximum amount that each copy can be rotated from the original, with 100 meaning plus or minus 180°; for Scale, a value of 100 means that the sprayed copies can be up to twice the size of the original. By setting both these to zero your copies will all be identical to one another – and to the original object, too. It's a quick and easy way to turn our simple leaves into a cartoon forest.

The final two sliders affect how the sprayed objects are distributed over the available area. Their effects are most visible when the



Width slider is quite large. Think of the Spray tool as placing copies of your object onto a circle: the Focus slider determines the size of the circle, and the Scatter slider determines how close to the circle each copy is placed.

Keeping the Scatter value low, it's easy to see the effect of the Focus slider. Setting it to 0 will keep all the copies in a tight circle under the cursor, regardless of the Width value. Putting it all the way to 100 will draw the objects around the periphery of the spray area, creating a ring of copies whose size is determined by the Width parameter. In this example, I've sprayed the same spot, with Scatter=0 but with different Focus values: the blue stars with Focus=0, green with Focus=20 and red with Focus=100.



With the Scatter control set to zero, the copies are placed very near to the circle that the Width and Focus controls define. Increasing the Scatter allows the copies to be placed further away from that ring – although they're still randomly positioned, so some will inevitably fall close to it. Putting it all the way to 100 gives Inkscape free rein to place objects anywhere within the spray area, at which point the Focus value ceases

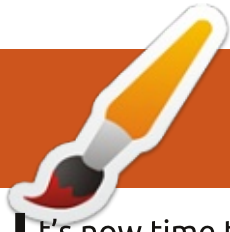
to have any real effect.

The Spray tool can be very useful for creating a random distribution of objects or, by constraining some of the values, a decorative path or outline. It's a shame that it's not possible to control more of the parameters using pressure or tilt on a graphics tablet. It's also unfortunate that the tool doesn't care about the direction of movement – it would be great to easily draw a line of footprints or arrows that twist and turn to follow your path as you move the mouse around. Despite those shortcomings, for certain effects the Spray tool is invaluable, and if you want to introduce some randomness into your drawings it's well worth exploring further.



Mark uses Inkscape to create three webcomics, 'The Greys', 'Monsters, Inked' and 'Elvie', which can all be found at

<http://www.peppertop.com/>



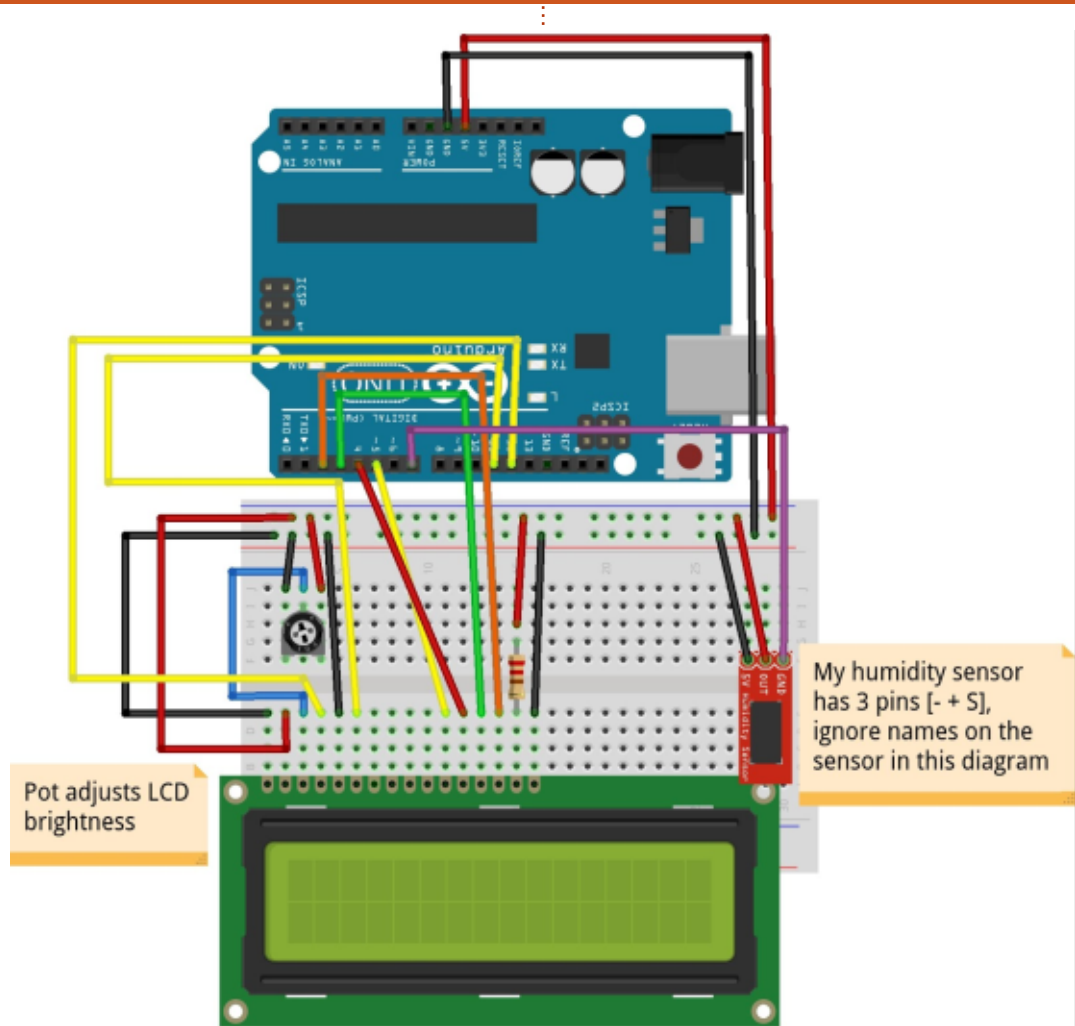
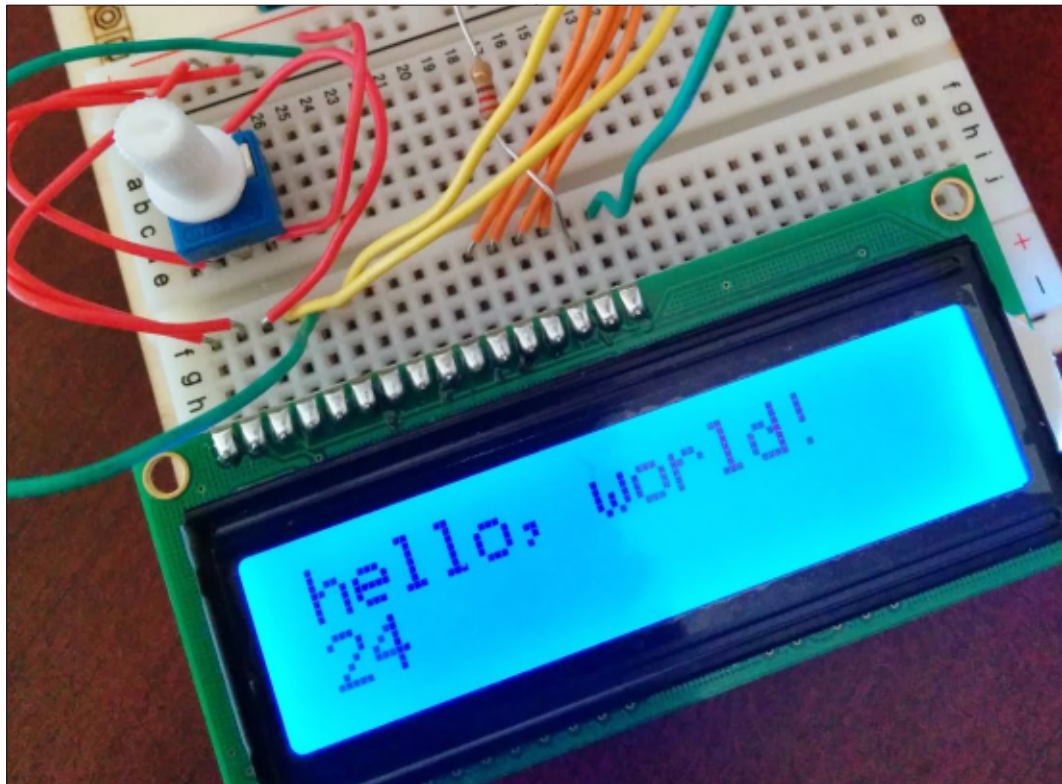
HOW-TO

Written by Ronnie Tucker

It's now time to break out the LCD screen and get some off-computer display going on.

I have to be honest and admit that it took me about five tries to get the LCD screen to work properly. The example code in the Arduino IDE explicitly states that the LCD needs only about 6 wires. Rubbish! It requires about 12 of the 16 used as it (obviously) requires

power, ground, and at least one wire to a potentiometer (pot) or PWM pin to control the screen brightness. After about two tries I was getting a bit frustrated and ended up just pulling all the wires out and starting again. Fifth time lucky! I finally got a "Hello World!" message and could brighten/darken the screen using a 10k pot.



fritzing

HUMIDITY SENSOR

I'd previously tinkered with my humidity sensor. That took a bit of investigating to get some decent code as the code with the sensor

kit wouldn't even compile. I eventually found some code here: <http://playground.arduino.cc/main/DHT11Lib> which is for the DHT11 module (as used by this sensor). As it says on that page, you need to

HOWTO - ARDUINO

create two files – dht11.h and dht11.cpp – and paste the relevant code (from that page) into the two files. Now I can compile the code from that page to get a working humidity sensor which displays the temperature and humidity to the serial display in the Arduino IDE. Wiring up the sensor was easy enough as it needs only 5v (middle pin, not labelled), - (ground), and the S(ensor) pin going to the Arduino.

With those two working (independently) it was time to try and combine the code (and parts) into one board, and create a digital thermometer. It was easier than I had expected. I opened the humidity sensor code and simply copied over the LCD include, initialisation line, and included the humidity variables in the lcd.print commands – and voila!

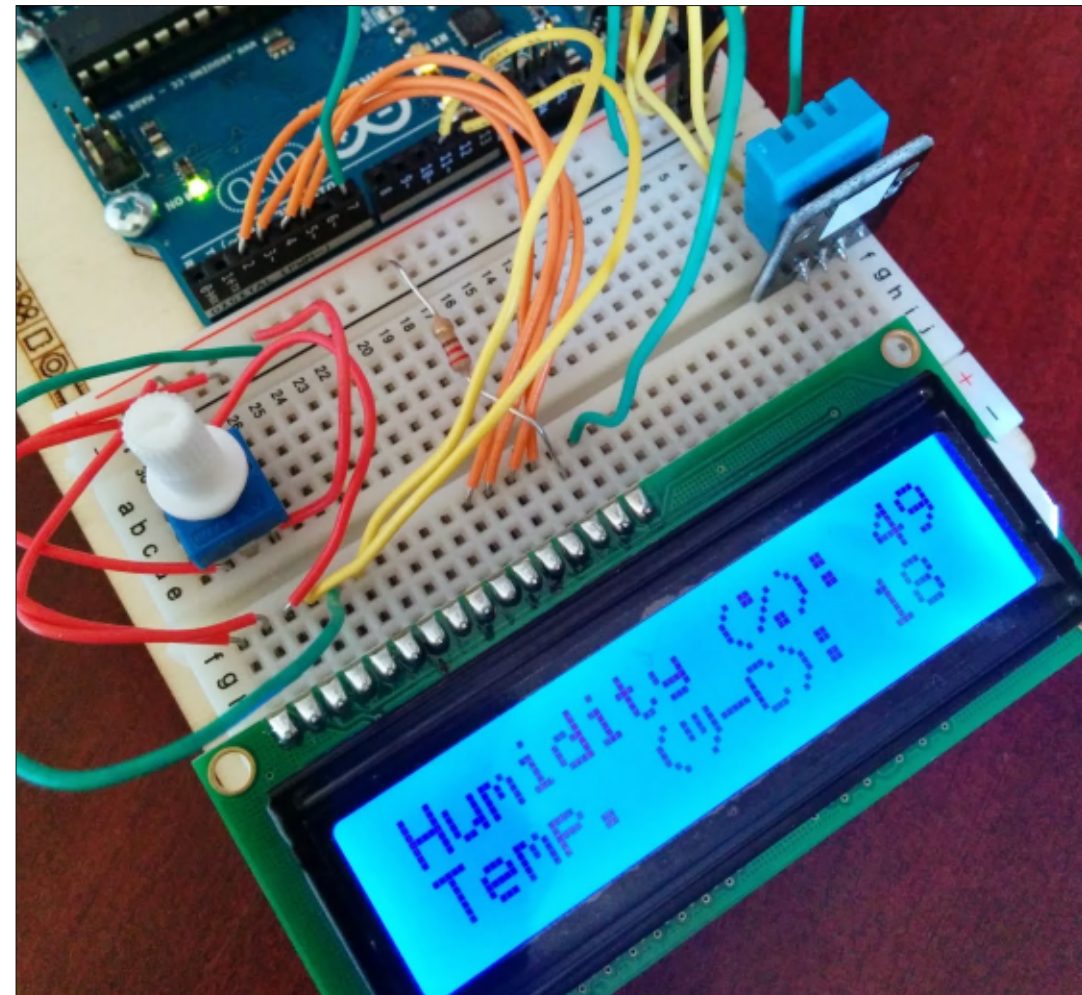
My code (for your enjoyment) is at: <http://pastebin.com/jtkK38ES>

ASSEMBLE!

```
// degree fahrenheit
byte degFChar[8] = {
  B01000,
  B10100,
  B01000,
  B00011,
  B00100,
  B00111,
  B00100,
  B00000
};

void setup()
{
  // ...
  lcd.createChar(3, degFChar);
  // ...
}

void loop()
{
  lcd.setCursor(0,0); //Start at character 0 on line 0
  lcd.print("Temp:");
  lcd.print(tempValue);
  lcd.write(byte(3)); //Degrees f
}
```



As you can see, the degrees Centigrade ('C) isn't displaying quite properly. A kind Redditor messaged me the code shown left (which I haven't tested as yet).

Here is a cool tool that allows you to create your own custom characters:

<http://fusion94.org/lcdchangen/>



Ronnie is the founder and (still!) editor of Full Circle. He's a part-time arts and crafts sort of guy, and now an Arduino tinkerer.





HOW-TO

Written by Ronnie Tucker

Write For Full Circle Magazine

GUIDELINES

The single rule for an article is that **it must somehow be linked to Ubuntu or one of the many derivatives of Ubuntu (Kubuntu, Xubuntu, Lubuntu, etc).**

RULES

• There is no word limit for articles, but be advised that long articles may be split across several issues.

• For advice, please refer to the **Official Full Circle Style Guide:**
<http://url.fullcirclemagazine.org/75d471>

• Write your article in whichever software you choose, I would recommend LibreOffice, but most importantly - **PLEASE SPELL AND GRAMMAR CHECK IT!**

• In your article, please indicate where you would like a particular image to be placed by indicating the image name in a new paragraph or by embedding the image in the ODT (Open Office)

document.

• Images should be JPG, no wider than 800 pixels, and use low compression.

• Do not use tables or any type of **bold** or *italic* formatting.

If you are writing a review, please follow these guidelines :

When you are ready to submit your article please email it to:
articles@fullcirclemagazine.org

TRANSLATIONS

If you would like to translate Full Circle into your native language please send an email to ronnie@fullcirclemagazine.org and we will either put you in touch with an existing team, or give you access to the raw text to translate from. With a completed PDF, you will be able to upload your file to the main Full Circle site.

REVIEWS

GAMES/APPLICATIONS

When reviewing games/applications please state clearly:

- title of the game
- who makes the game
- is it free, or a paid download?
- where to get it from (give download/homepage URL)
- is it Linux native, or did you use Wine?
- your marks out of five
- a summary with positive and negative points

HARDWARE

When reviewing hardware please state clearly:

- make and model of the hardware
- what category would you put this hardware into?
- any glitches that you may have had while using the hardware?
- easy to get the hardware working in Linux?
- did you have to use Windows drivers?
- marks out of five
- a summary with positive and negative points

You don't need to be an expert to write an article - write about the games, applications and hardware that you use every day.





Online
BACKUP

Secure
SYNC

Easy
SHARING

Whether you need to access a document you have stored on a remote server, synchronize data between a Mac, Windows or Linux device, share important business documents with your clients, or just rest easy knowing all of your data is safely, securely, and automatically backed up - SpiderOak's free online backup, online sync and online sharing solution can handle all your needs!

SpiderOak offers a different approach to online backup by combining a suite of services into one consolidated tool - free online backup, synchronization, sharing, remote access, and storage. This difference is further measured in our zero-knowledge privacy policy - the first one ever employed in this setting. Our flexible design allows you to handle data from any operating system (Mac, Windows and Linux) or location (external drives, network volumes, USB keys, etc...) using just one centralized account.

- ➔ Access all your data in one de-duplicated location
- ➔ Configurable multi-platform synchronization
- ➔ Preserve all historical versions & deleted files
- ➔ Share folders instantly in web ShareRooms w / RSS
- ➔ Retrieve files from any internet-connected device
- ➔ Comprehensive 'zero-knowledge' data encryption
- ➔ 2 GBs Free / \$10 per 100 GBs / Unlimited devices

<https://spideroak.com>

Download mobile clients
for **iOS & Android**

JOIN SPIDEROAK NOW
Get 2 Free GBs

Get 25% off any SpiderOak package
with the code: **FullcirclemagFans**



If you follow FCM on Facebook, Twitter, Google+ or you're on our mailing list you'll have read the harrowing account of my great loss. A grand total of 1TB of hard drive data, years' worth, including the Scribus files for FCM#79 (hence why it looked half done, it was!). As an experiment I decided to try and recover some data just to see if it was possible and what I'd get.

I'd like to thank all of you who emailed me support and links back then to recovery software.

THE BACKGROUND

So, in short: FreeNAS formatted and repartitioned my hard drive then installed itself on a 4GB partition leaving almost 1TB of unallocated space just hanging there.

THE SOLUTION

I grabbed a magazine DVD and installed Mint 15. I took back my hard drive by removing the

partitions created by FreeNAS and created a 150GB partition (root) an 800GB partition (/home) and a 50GB partition (swap - on the off chance that I might need it). All of them were formatted and Mint 15 installed.

THE SOLUTION AGAIN

Since Mint 15 was giving me some teething problems with dual monitors and display settings I reformatted the root partition and installed Mint 16 RC. This was much better and is the OS I'm still using now.

MORBID CURIOSITY

So, having installed Back In Time for an automated backup (horse, door, bolted) I began to wonder what data I could recover, if any. I'd accepted that it was gone for good so whatever I could recover was just for curiosity sakes.

PHOTOREC

I decided to try Photorec since it

was the most widely recommended. You can either install it from the Photorec site (<http://www.cgsecurity.org/>) or from your distro package/software manager.

Photorec is a command line app that is run by using the command:

```
sudo photorec
```

You are then presented with some options:

```
PhotoRec 6.14, Data Recovery Utility, July 2
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

PhotoRec is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press E)
>Disk /dev/sda - 1000 GB / 931 GiB (R0) - WD
Disk /dev/sdb - 122 GB / 114 GiB (R0) - Max
Disk /dev/sdc - 60 GB / 55 GiB (R0) - ST360

[Proceed] [Quit]
```

I went with the default settings and chose my 1TB drive (/dev/sda). Then you choose a partition from that drive to work with:

```
PhotoRec 6.14, Data Recovery Utility
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

Disk /dev/sda - 1000 GB / 931 GiB

Partition
No partition 0
1 * Linux 0
2 E extended 18236
X extended 18236
> 6 L Linux Swap 18236
5 L Linux 24315
```

For my experiment I chose what is now my 50GB swap drive. Next, you choose which file system the lost files were on:

```
PhotoRec 6.14, Data Recovery Utility
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

6 L Linux Swap 18236

To recover lost files, PhotoRec will
file were stored:
>[ ext2/ext3 ] ext2/ext3/ext4 file
[ Other ] FAT/NTFS/HFS+/ReiserFS
```

Next, you choose (using the arrow keys) where you'd like the recovered files to go:


```
PhotoRec 6.14, Data Recovery Utility, July 2013

Please select a destination to save the recovered files.
Do not choose to write the files to the same partition they were s
Keys: Arrow keys to select another directory
      C when the destination is correct
      Q to quit
Directory /media/ronnie
drwxr-x---  0  0      4096 25-Nov-2013 22:11 .
drwxr-xr-x  0  0      4096 24-Nov-2013 12:14 ..
>drwxr-xr-x 1000 1000   4096 25-Nov-2013 22:11 Backup
```

Now you let Photorec run its course.

Photorec took around 25 minutes to scan the 50GB partition in my example and, surprisingly, came back with over 5,000 recovered files! Granted, not all were 100% recovered. Some video files were only one quarter their original length.

CONCLUSION

While 5,000 files sounds great, and it is for a double repartition and reformat, a lot of that was junk from my browser history and there's no way to know what a file is until you open it. All files are renamed f0000000.xxx where 0000000 is a number and .xxx is the

file extension. So don't expect to get back your directory structure and files by their original name.

Still, I was impressed that I could get anything back let alone 5,000 files.

I never did try that 800GB partition...

More Photorec info:
http://www.cgsecurity.org/wiki/PhotoRec_Step_By_Step



Ronnie is the founder and (still!) editor of Full Circle. He's a part-time arts and crafts sort of guy, and now an Arduino tinkerer.

LIBREOFFICE SPECIAL EDITIONS:



<http://fullcirclemagazine.org/libreoffice-special-edition-volume-one/>



<http://fullcirclemagazine.org/libreoffice-special-edition-volume-two/>



<http://fullcirclemagazine.org/libreoffice-special-edition-volume-three/>

INKSCAPE SPECIAL EDITIONS:



<http://fullcirclemagazine.org/inkscape-special-edition-volume-one/>



<http://fullcirclemagazine.org/inkscape-special-edition-volume-two/>



MY STORY

Written by T.Kovács Áron

My Start With Ubuntu

I have been interested in informatics since I was a child. I started in those ancient times when there was a blue image on the monitor and we had to load the games from a tape. After a while I started learning programming at the school, and I wrote small programs in Pascal language. It was one of my favorite periods of my life, but I quit and never got to C+. The usual Operating System then was DOS, with Norton Commander file-manager and text-based surfing on the internet.

I bought my first computer when I studied at the University. Obviously, I was using Windows operating system like everyone else. There was only one odd guy in the next building, using Linux. Windows 2000 was pretty good but I had to reinstall it every half year. I tried XP, but my computer was too weak for it so I turned back to the previous version every time. After a while I started to work, and I had a list of my favorite programs. I started to think that I didn't want to change everything—new computer, new system, new

programs—every one and a half years. So I made a change in my computer usage.

I started to search for new possibilities and tried Linux. It was Mandrake and I got it on two CDs. But I didn't like it and I couldn't manage to work with it. I was searching again for several years. I've found a "human-friend" Linux distribution: Ubuntu. I think it was the 8.04 version that I first downloaded and installed alongside the Windows system. I loved it and I started to discover

and learn that it was straight, practical, and very fast. Unfortunately I wasn't able to put aside the Windows system in my daily work because of the programs we were using. Architectural designing software doesn't run on Linux up to this day...

We had an old computer in the office. We installed Ubuntu on it and put it in the corner. This was our "server" and the only computer with internet access. So we resolved three problems at once: 1)

we had no problem at all with viruses, 2) we didn't spend time on the internet during working hours, and 3) we didn't have to throw out an out-of-date computer. This success has convinced me about the justification of Linux. I have decided to use Linux in the long run.

After a while, I started my own business and the problem was getting serious. The question was whether I should buy a new computer with a new OS and the programs I was using in order to continue my work, or if I should solve all this with the help of Linux. I chose the second way, and I again started to search for Linux software that could help me in my work as an architect.

First I gave up CAD designing and chose to draw by hand which was more appropriate for me. The elaboration had to be done by computer: scanning, working with images, document assembling, written parts, tables, and printing. Under Ubuntu 10.04, I used OpenOffice, Xsane, Scribus, Gimp

```
Turbo Pascal
File Edit Search Run Compile Debug Options Window Help
COUNT1.PAS
Error 3: Unknown identifier.
var
  invalid_operator : boolean;
  operator : char;
  number1, number2, result : real;
begin
  invalid_operator := FALSE;
  writeIn('Enter two numbers and an operator in the format');
  writeIn(' number1 operator number2');
  readIn(number1); readIn(operator); ReadIn(number2);
  if operator = '*' then result := number1 * number2
  else if operator = '/' then result := number1 / number2
  else if operator = '+' then result := number1 + number2
  else if operator = '-' then result := number1 - number2
  else invalid_operator := TRUE;

  if invalid_operator then
    writeIn('Invalid operator')
  else
    writeIn(number1:4:2, ' ', operator, ' ', number2:4:2, ' is ',
      result:5:2)
end.
7:17
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
```

and Inkscape software. LibreCad and Qcad programs seemed suitable in collaborating with my partners (for reading and editing DXF files).

After that I discovered the DraftSight software, which helped me to resolve this question. Even though I didn't like the Autocad cloned programs, I didn't find a suitable, intuitive and free 2D drawing program. I managed to get used to it as much as I needed to, and I am still using it constantly. I also tried 3D modeling programs, but the Blender is too complicated, the FreeCad is hard to use, the other ones are not so suitable for me.

A new version of Linux was introduced to me by one of my friends. He was using Linux Mint so I tried it. Ubuntu had just changed to the Unity interface, which I disliked. Linux Mint, on the other hand, brought me exactly what I needed – the old familiar usability and all the things I liked in Ubuntu, plus a new range of free programs to choose from.

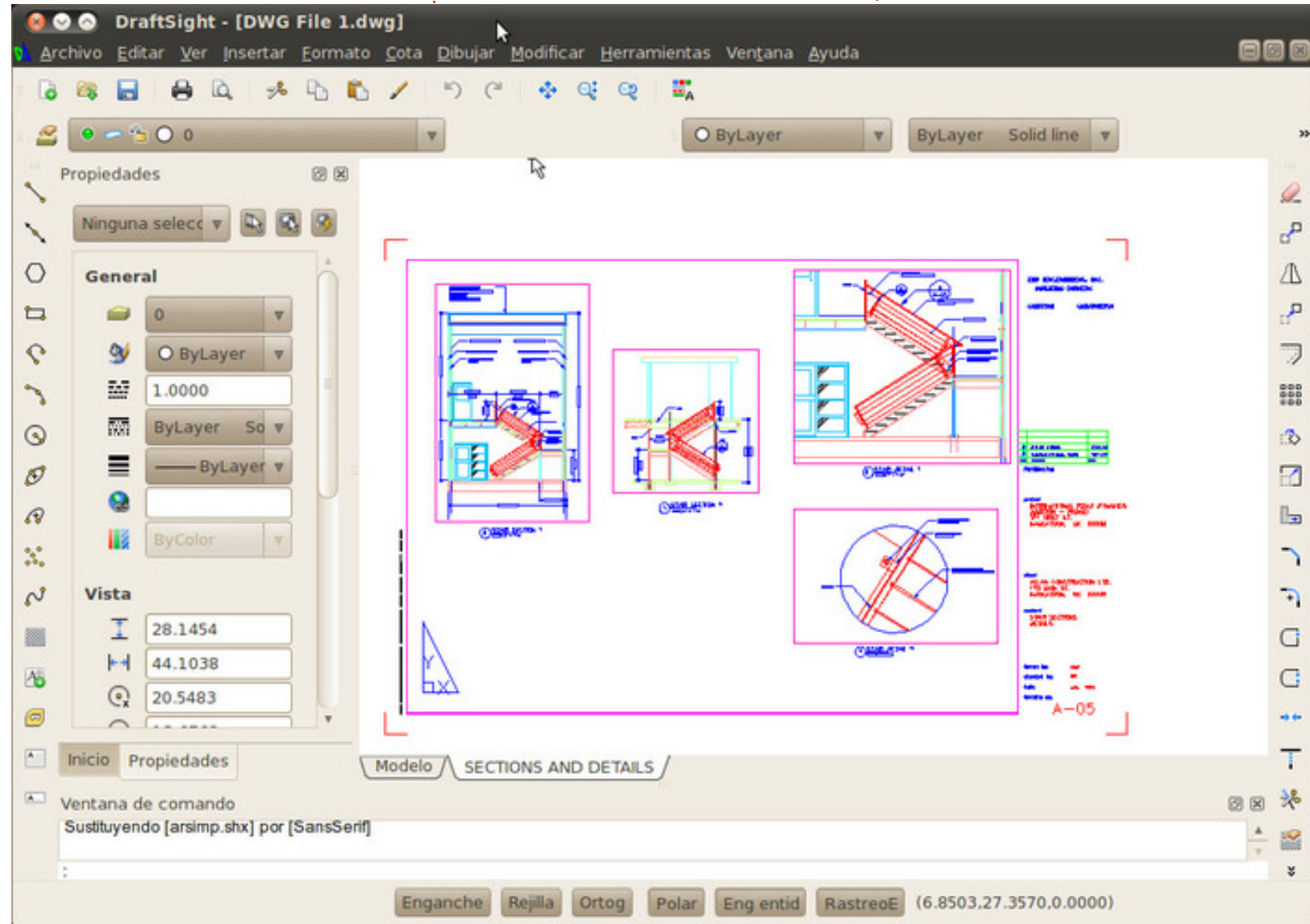
I have continued to use Linux Mint in the last several years. My only dilemma was the desktop

manager: I had to choose from Cinnamon/Mate/KDE/XFCE. I changed my system many times trying different versions.

In the meantime, I have made a lot of discoveries and found

solutions for many problems concerning software and hardware. And I've started to like the Open idea more and more and started to recommend this possibility to others too. I managed to convince more people to try Linux instead of

Windows. Some of them have continued to use it. Even today I am happy to recommend it if somebody is having trouble with his computer. One simple solution: put in a Live CD and it's already running!





Toshiba MQ01ABD100H SSHD

This hard drive is a 2.5" unit for laptops, that, at approximately 8mm in height, should fit most laptops and netbooks. It combines a standard platter 1000 GigaByte hard drive, and an 8 GigaByte Solid State Drive (SSD), both integrated into a hybrid unit that Toshiba calls a Solid State Hybrid Drive (SSHD). The SSD part is a smaller but much faster storage space, and is used as a cache to access data at higher speeds than would normally be possible with a platter drive. The cost is lower than would be possible with a pure SSD drive of similar capacity, though higher than a platter drive. Data management is performed by the hard drive circuitry, with no intervention needed (nor indeed possible) from the operating system. This is a departure from Apple's Fusion Drive, where the OS itself manages data transfers between the SSD and the platter parts of the drive.

I tested this unit in an Acer Aspire AO-722. This 11.6" netbook has an AMD C-60 64-bit processor, and originally came with a 320

GigaByte platter drive. When upgrading to an SSD drive in search of more speed, the small case dimensions meant that the platter drive had to go to make space for the replacement SSD drive, a Crucial M4. This worked fine, system and application boot times went way down and performance

was in line with what could be expected from a lightweight computer with a fast drive: the limiting factor was now the processor, not the hard drive. However, I had achieved this at the expense of losing disk space, since the Crucial unit holds only 64 GB. Other SSD drives were available,

but at extra expense. Though their prices have gone down during the last year, users should still expect to pay about \$1.00 per GigaByte: large SSDs in excess of 500 Gigabytes may be worth more than the computer itself! If the hybrid drive holds its promises, I may be able to get the best of both



technologies within the limited physical space available for this small computer.

I already had a working Xubuntu 14.04 system I was happy with, so I decided to clone the existing setup instead of going through the full installation process. The system detected the new drive - connected through an external USB adapter - as a single unit with no problems. The drive comes completely uninitialized, with no partition table as reported by gparted.

My original partitioning consisted of an ext4 boot partition. The rest of the Crucial drive was set up as an LVM physical unit, out of which I had carved a 15 GigaByte logical volume for the system root, and another for /home. There was still some space available for future applications.

```
/dev/sda2
243M  40M  187M  18% /boot
/dev/mapper/SSD--VG-System
15G   7,8G   7,1G  49% /
/dev/mapper/SSD--VG-Home
20G   5,0G   14G   27% /home
```

After creating an MS-DOS partition table and partitioning the new Toshiba hybrid drive in the

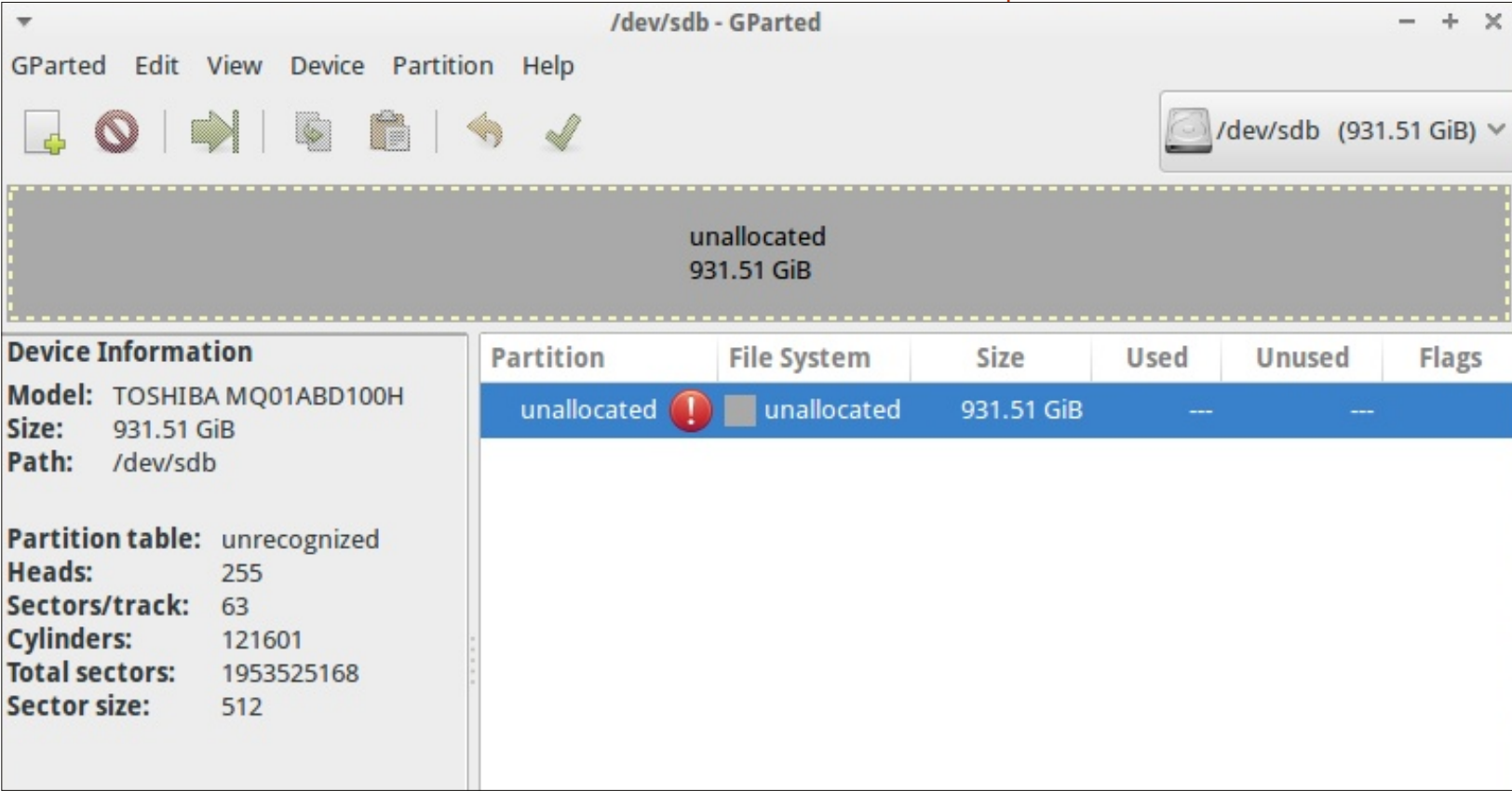
same way, I then copied over each partition, installed GRUB on the new unit, and booted the computer from the new drive over USB just to make sure everything was working correctly. I now got up to 901 GigaBytes free for user data - or 850 GB when the standard 5% was reserved for root's use.

```
/dev/sda1          976M
40M   870M   5% /boot
/dev/mapper/SSHD-System 15G
7,8G   7,1G  49% /
/dev/mapper/SSHD-Home  901G
5,0G   850G   1% /home
```

I then switched off the computer and got out the old screwdriver to install the hard drive in its place. The hard drive is at the top left of the picture, with the CPU and its cooling fan visible at the top right, RAM slots at bottom right and the WiFi card at bottom left. As you can see, there is little space left over in this computer!

Now, for some testing. I compared several typical actions

both with the former Crucial SSD and the new hybrid drive. In both cases, the system comes up in 31s - there are no measurable differences. With the new drive, Gimp starts up in 18s, while LibreOffice Writer needs just 5s. These are just about the same times measured with the SSD, and a definite advance over the traditional spinning disk drive this computer came with. Speeds are much higher and the system is much more responsive. In fact, I did



REVIEW - TOSHIBA SSHD

not see any user-noticeable differences between the hybrid Toshiba drive and the pure SSD drive - at least, not during everyday tasks.

From a technical standpoint, there are some limits. The hybrid drive has the same SATA-III 6 Gbps interface most SSD drives have today. However, for the time being, no consumer hard drive technologies will fill this bus up completely: laptop platter disks spinning at 5400 rpm are limited in real terms to read speeds in the 100 - 120 MByte/s range, while SSDs may get up to 300-400 MByte/s. As for the hybrid drive, it has been clocked at up to 172 MByte/s read speed

([http://hdd.userbenchmark.com/To shiba-Notebook-SSHD-1TB/](http://hdd.userbenchmark.com/To shiba-Notebook-SSHD-1TB/Rating/1957&tab=Benchmarks)

[Rating/1957&tab=Benchmarks](http://hdd.userbenchmark.com/To shiba-Notebook-SSHD-1TB/Rating/1957&tab=Benchmarks)). However, it should be noted that results will depend on whether the data accessed is inside the SSD part, or if it needs to be retrieved from the platter. With this type of cache, we can expect best results from usage patterns that access small amounts of data that fit into the SSD part.

If we need to access large amounts of varied data such as in

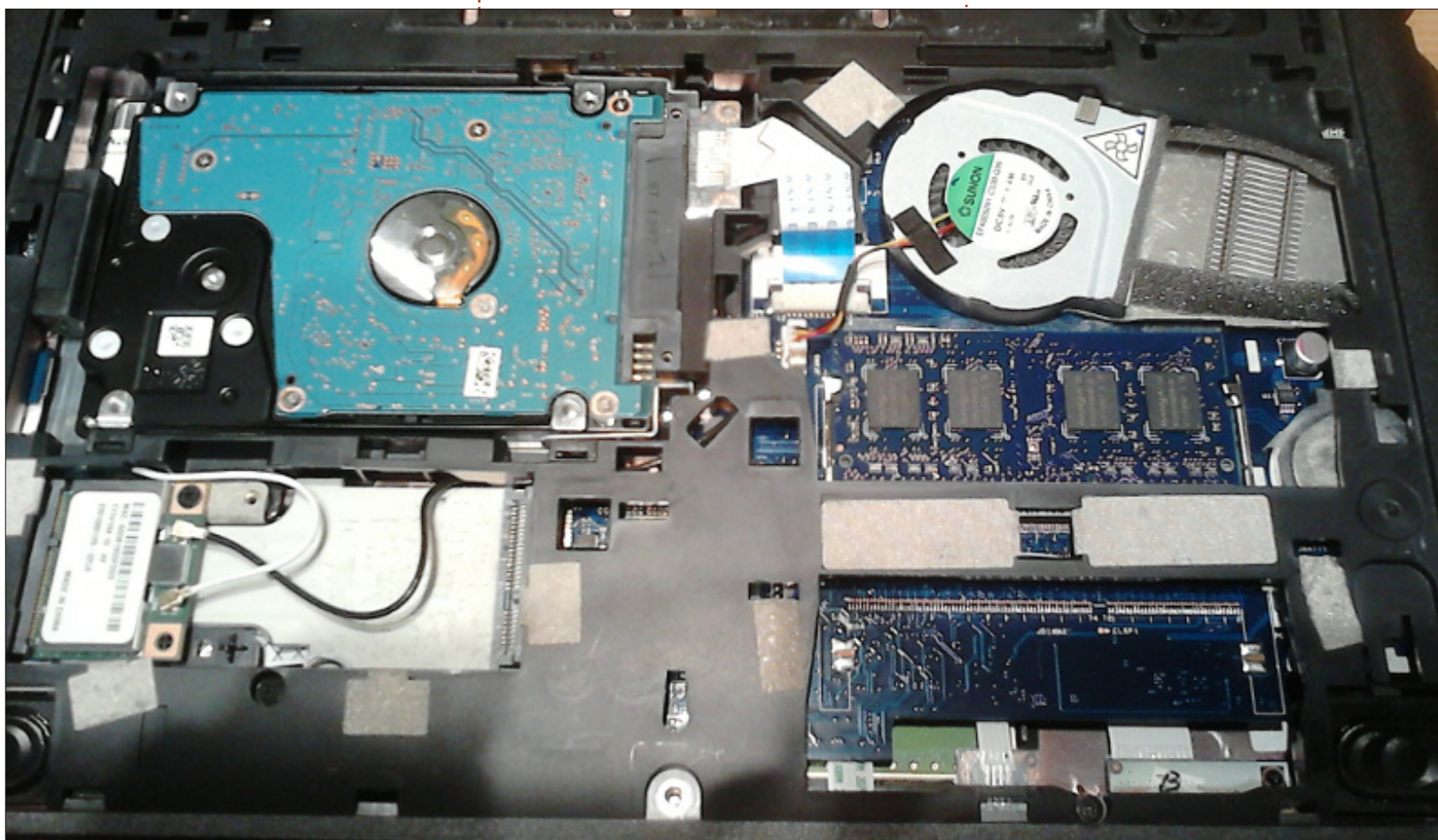
video editing, we could expect much of this data to reside on the slower platter, thus negating the usefulness of the hybrid drive concept. On the other hand, a small, compact, operating system used for Internet access and light office tasks is ideal - and this is just about the projected use of a netbook with Xubuntu. Most system applications and user data fit within the 8 GigaByte cache and

are accessed at SSD speeds. Other, larger and less-often accessed data stay within the 1 TeraByte platter and are accessed when necessary, though at a slower pace.

All in all, this concept of hybrid drive is probably a very pertinent upgrade for your netbook, though perhaps less so for a tower computer that could fit in an extra SSD as well as the original disk.



Alan teaches computer science at Escola Andorrana de Batxillerat (high-school). He has given GNU/Linux courses at the University of Andorra and currently teaches GNU/Linux systems administration at the Open University of Catalunya (UOC).





WHAT IS

Written by Ronnie Tucker

As Oscar explained in the previous issue (FCM#85) there are many different types of Bitcoin alternatives, or as they're sometimes known: alt-coins. The most popular, I think, is the Dogecoin – they've done some very high profile marketing such as the Jamaican Bobsled Team and the more recent Josh Wise NASCAR race. So, let's take the Dogecoin wallet source and compile it.

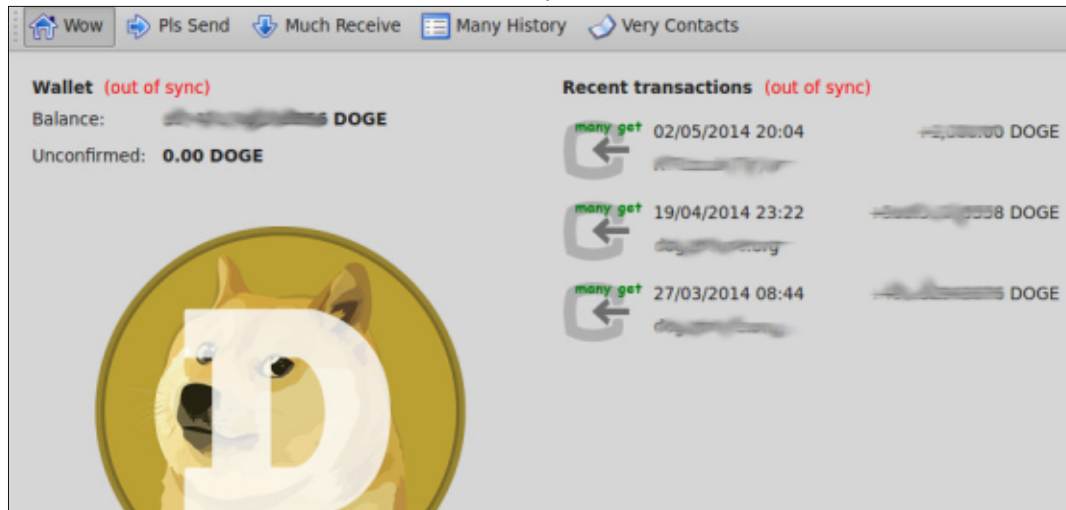
Although this article is discussing the Dogecoin wallet, this procedure can be used for the vast majority of alt-coin wallets.

First, we need the tools to compile the source. In a terminal type (all one line):

```
sudo apt-get install libssl-dev libdb-dev libdb++-dev libqrencode-dev qt4-qmake libqtgui4 libqt4-dev libminiupnpc-dev libminiupnpc8 libboost-all-dev build-essential git
```

GET GIT

Now, we need some source to compile. In the same terminal type:



```
git clone https://github.com/dogecoin/dogecoin.git
```

What we've done there is copy the Dogecoin wallet source to our machine at /home/dogecoin. Next, we need to enter the Dogecoin wallet folder. So, type:

```
cd dogecoin
```

Now we're in the right place; we need to run a global-modify (-i = in-place) of the source. Type:

```
sed -i 's/-mgw46-mt-sd-1_53//g' dogecoin-qt.pro
```

MAKE

Compiling An Alt-coin Wallet

The wallet, on first run, will need to sync with the Dogecoin network and it will take a while. I'm talking many hours. It effectively has to download a ledger of every transaction. That's a lot.

If you were to compile the Reddcoin wallet, then you'd simply replace all instances of 'dogecoin' (in the above commands) with 'reddcoin'. Same with Litecoin. It's that simple. For example:

```
git clone https://github.com/dogecoin/dogecoin.git
```

would become:

```
git clone https://github.com/reddcoin/reddcoin.git
```

and so on.

Next issue I'll discuss the dualminer USB widget I bought a couple of months ago, and show how I finally managed to get it mining in Linux.

To the moon!

We can finally make something now. Type:

```
qmake USE_UPNP=- USE_QRCODE=0 USE_IPV6=0
```

When that's complete, type:

```
make -j3
```

And when that's done, we're done. To run the Dogecoin wallet you type:

```
./dogecoin-qt
```

Or you can use your desktop environment to double-click the dogecoin-qt icon.



LETTERS

If you would like to submit a letter for publication, compliment or complaint, please email it to: letters@fullcirclemagazine.org. PLEASE NOTE: some letters may be edited for space.

LIBREOFFICE SPELL CHECKER

I use Xubuntu, and have done for some time now. As 14.04 was an LTS version, I decided to do a fresh install and I am pleased with it for the most part. I installed LibreOffice by opening a terminal window and typing `sudo apt-get install libreoffice`. So far, so good. LibreOffice installed quickly. I had used it for a time without noticing that the spell-checker did not work. Some quick checks did not show anything amiss, so I did a bit of internet searching and found that the `myspell-en-gb` dictionary was not installed. In Synaptic Package Manager, just search for `myspell` and scroll to the `en-gb` version. In Ubuntu Software Center and if you search for exactly `myspell-en-gb`, it will show up. Install the package and spell-check is restored. I hope that it saves some time and frustration for someone.

Iain Mckeand

BEWARE THE ROBOLINUX

Regarding your first news item in Issue 85 of Full Circle, I would like to warn Ubuntu users, as someone who attempted to use this since April, that Robolinux is not really ready for prime time.

Robolinux claims to be a Debian distribution; it is one of the ugliest repackagings of Linux I have seen, and lacks much that Ubuntu has. A replacement for Ubuntu it is not. Its really unique claim to fame is that it can allegedly re-package an installed Windows XP partition, system and applications, into a Virtualbox virtual machine, which one can presumably run under Ubuntu.

This may be true if you do not dual-boot Windows and Ubuntu. I wish they had stated this up-front, when I told them I dual-boot. If you are like many regular Ubuntu users, who dual-booted XP under GRUB as a security blanket, you're out of luck. Robolinux won't work.

In April, Robo was a series of

scripts that implanted itself on the Ubuntu Gnome menu. That had many errors that I pointed out as a courtesy, believing the project to be worthwhile. In May, a new package was released.

The new version consists of several borrowed Sysinternals utilities, for the packaging of the partition within Windows. Then, on the Ubuntu side, the old scripts are wrapped into a `.deb` executable. A Robolinux message asserted that I had to remove GRUB and repair the MBR. I asked if there were simpler fixes than removing the MBR and GRUB

(I feared being locked out of both Windows and Ubuntu, with an as yet unproven script). That's when Robolinux declared I was "in violation of our Terms of Use," refunded the "donation," and refused further contact.

I do and have run other virtual machines, several Windows, and even a DOS, so I know my setup can run a properly assembled and installed machine. I liked the idea of not having to reinstall all my

Join us on:



goo.gl/FRTMI



facebook.com/fullcirclemagazine



twitter.com/#!/fullcirclemag



linkedin.com/company/full-circle-magazine



ubuntuforums.org/forumdisplay.php?f=270

FULL CIRCLE NEEDS You!



Without reader input **Full Circle** would be an empty PDF file (which I don't think many people would find particularly interesting). We are always looking for articles, reviews, anything! Even small things like letters and desktop screens help fill the magazine.

See the article **Writing for Full Circle** in this issue to read our basic guidelines.

Have a look at the last page of any issue to get the details of where to send your contributions.



Windows software (some of which I still need).

Robolinux is a promising idea, but it needs some careful development --- and perhaps new leadership.

The founder, John Martinson, apparently mistook my handle as female. He seemed more than a little enamored of the female user I was in his mind, and "dear"-ed me to death in a series of emails. I am of the male persuasion, always have been. His error is the only explanation I can find for a very odd email to me complaining that I didn't value him or his work, all with a level of emotion I thought was very inappropriate for the essentially technical messages we were exchanging.

Indeed, Robolinux may well be a one-man-band behind the project. One gets messages from a variety of "managers" and "customer service" people, but all messages come from one and the same email address. I have no problems with an entrepreneurial programmer (but why go with "donations" to aid alleged Cambodian children in exchange for the software?) until the one man behaves oddly toward

"customers" and their "donations."

Cecilieaux Bois de Murier

DOWNLOAD ALL THE ISSUES

After losing my backup I found it terribly inconvenient to download all issues of FCM via the web. if you think anyone would be interested, here is a simple script to download all FCM issues using a Python script (below right).

You copy/paste (or type) the text into a text editor, save the file to a directory of your choice, then make it executable, and run it.

In your favorite file manager, open the properties of this newly saved file and mark it as executable (normally on the permissions tab). As the script does not come with a user interface, you will see nothing when you execute it from your file manager so I recommend you use a terminal window to execute the script. Open a terminal window, change to the folder where you saved the file, such as:

```
cd ~/Downloads
```

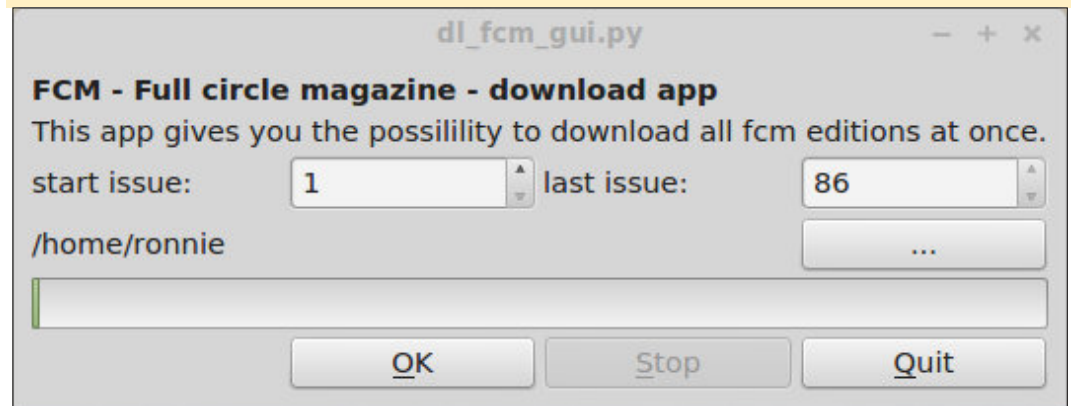
SCRIPT TO DOWNLOAD ALL ISSUES OF FCM

```
import urllib.request

#download number of issues to current directory

#sample url http://dl.fullcirclemagazine.org/issue85_en.pdf
path = ''
server = 'http://dl.fullcirclemagazine.org/'
fname_pre = 'issue'
fname_post = '_en.pdf'
num_start = 2
num_issues = 85

for i in range(num_start, num_issues + 1):
    fname = fname_pre + str(i) + fname_post
    url = server + fname
    print('Downloading ' + url + ' ...')
    urllib.request.urlretrieve(url, path + fname)
```



or grab the GUI version using:

```
wget
www.liedler.at/dl/dl_fcm_gui.py
```

whichever one you get, mark it as executable with:

```
chmod +x dl_fcm.py
```

or:

```
chmod +x dl_fcm_gui.py
```

and either double click the GUI file, or run the script with:

```
./df_fcm.py
```

Peter Liedler





Q&A

Compiled by Gord Campbell

If you have a Linux question, email it to: misc@fullcirclemagazine.org, and Gord will answer them in a future issue. Please include as much information as you can about your query.

Q How do I get rid of the annoying Grub boot choice screen in the latest Ubuntu 14.04 and make it directly go to the logon screen?

A There's a good reason to keep the Grub screen: if you install a new kernel, and it doesn't work on your system, Grub lets you boot with an older kernel.

If you think Grub is slowing things down, just press Enter.

Q When I print from Firefox, the background images do not appear.

A (Thanks to **Barry** in the Yahoo UbuntuLinux group) Press Ctrl-P, select the Options tab, click on "Print Background Images".

Q I'm currently working on Ubuntu installed on a flash drive, through the tryubuntu option. However, every time I reboot, the files which I've saved don't come back.

A (Thanks to **sudodus** in the Ubuntu Forums) You can make a persistent USB boot drive with the 'Startup Disk Creator' (alias usb-creator) or with Unetbootin. These programs offer the option to create a storage space for persistence, where you can save updates, new programs, tweaks, documents, pictures, etc.

If you are creating the USB boot drive from Windows, try Pendrivelinux.

Q In Ubuntu 14.04, I use a journal program called Lifeograph which I installed from the Software Center. It appeared in my apps list and I was easily able to add it to my dock. Trying the same thing using the latest Mint Debian with Cinnamon desktop, I install the app from their version of software center, but cannot find it on my system.

A It showed up under Accessories for me. The quickest way to find files is by using the Locate command, but

new files probably won't appear.

Q I am always running into some Pulse Audio quirk like muting when starting up Audacity. Can I not just uninstall Pulseaudio and use Alsa mixer? If so, how would I go about doing this? Are there any issues I should be aware of?

A (Thanks to **Temujin** in the Ubuntu Forums) It's technically possible, but you're probably better off starting with a distro that doesn't use it by default. (Eg. Lubuntu).

Q Does anyone know of any reasonably cheap 7" tablets that come with ubuntu preinstalled?

A Ubuntu Touch has been slow to gain a foothold. Just this month, Dell announced a tablet running Ubuntu for \$450. (Each Dell country manager picks what products will be available in his or her country, so you might not have access to it.) I have not seen any

previous announcement of a tablet with Ubuntu Touch, but it's not something I monitor.

You can install Ubuntu Touch on a Nexus 7, which runs about \$250. The cheaper tablets probably don't meet the minimum specs for Ubuntu Touch. (Eg. Toshiba Excite 7-inch, \$100.)

For lowest cost, go with a netbook.

Q I have got Windows7 running in Virtualbox on Ubuntu 14.04 solely so I don't have to keep changing drives for iTunes. Do I need to look at getting internet security?

A Yes, install Microsoft Security Essentials and keep it up to date.



TOP NEW QUESTIONS AT ASKUBUNTU

* A batch lyric fetcher and store in music files?

<http://goo.gl/PUOA8Q>

* Why do commented lines in a linux configuration file sometimes work?

<http://goo.gl/ThOAit>

* Where should I put my script so that I can run it by a direct command?

<http://goo.gl/JrrQu4>

* Run last command with all the letters transformed to lowercase?

<http://goo.gl/kDNQLf>

* How can I ensure a service is disabled on boot?

<http://goo.gl/9pRcbW>

* How to find the creation time of a file?

<http://goo.gl/EI7cr6>

* CPU Temperature unit?

<http://goo.gl/OCLso5>

* What's the difference if I use Windows 7 as a Standard User vs if

I use Ubuntu?

<http://goo.gl/mVbAKd>

* Renaming hundreds of files at once for proper sorting?

<http://goo.gl/zSB20j>

TIPS AND TECHNIQUES



Recycling computers

My office has replaced numerous old computers running Windows XP, with shiny new computers. What to do with the old machines?

The first priority is to ensure no data is lost, so we install Macrium Reflect on the old computers, and create an "image" of each old computer's hard drive. This goes onto an external drive, then is copied onto the user's new computer, which has a much larger drive. We also have a central system with copies of all the images. When the external drive fills up, we don't delete files, we replace it.

We're very nervous about confidential files being recovered from the old hard drives, so we

boot Darik's Boot and Nuke from a CD, and run it to completely erase the old hard drive.

Now we can install Linux. One choice is 32-bit Linux Mint 17 with Mate. These are old machines, after all.

Finally, we can give the computers away. In Toronto, we have a FreeGeek project which will be delighted to get fully functional computers. All done!



Gord had a long career in the computer industry, then retired for several years. More recently, he somehow found himself "The IT Guy" at a 15-person accounting firm in downtown Toronto.

Full Circle Podcast Episode 41, Trusted To Fail!!

Welcome to our new format show, there are several changes from the previous format, the most important being we are now recording together at the Blackpool Makerspace in the office. This Episode we Test Ubuntu 14.04, Review of Official Ubuntu Server Book.

Your hosts:

- Les Pounder
- Tony Hughes
- Oliver Clark



from the
Blackpool (UK) LUG
<http://blackpool.lug.org.uk>

Download



Security

Written by Kevin O'Brien

If you have security-related questions, email them to: misc@fullcirclemagazine.org, and Michael will answer them in a future issue. Please include as much information as you can about your query.

In the last few weeks (as I write this in late April 2014) two events have combined to deliver a powerful lesson on the security of Open Source software. But it is important to know exactly what the right lesson is. I have seen reports that Heartbleed was a proof of something fundamentally wrong with the Open Source model, because it denied the accuracy of Eric Raymond's famous saying "With many eyeballs all bugs are shallow." The Heartbleed bug was in a significant number of systems (actually about one-sixth of Internet sites, as far as I can tell from an analysis of how many sites use OpenSSL, and what percent of those use the versions of the software that are affected). There was a bit of hyperbole in how bad it was, but it is no doubt still pretty bad. But how did that happen?

I will refer everyone to an excellent article that has all of the details. It is called How Did the Heartbleed OpenSSL Bug Happen? (<http://www.digitaltrends.com/computing/how-did-the-heartbleed-openssl-bug-happen/#!FLdxR>), and

I recommend looking at it. It is short and to the point. Basically, there was a request to have an extension to OpenSSL to provide something called a TLS Heartbeat extension. This is a perfectly reasonable thing to do, and is covered in RFC 6520, Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension (<https://tools.ietf.org/html/rfc6520>). As the RFC makes clear, the purpose is to provide a "keep alive" functionality without requiring a renegotiation. OpenSSL was just trying to be compliant in adding a capability that the Internet Engineering Task Force had decided should be provided. But how does the OpenSSL project handle this?

The first thing we notice is that OpenSSL has a core team of just 11 people, most of them volunteers, and only one full-time person devoted to the project. Generally, they get about \$2000 a year in donations, and make some money from support contracts. In other words, they are stretched tight. A

volunteer in Germany, Dr. Robin Seggelmann, wrote the code to implement RFC and submitted it for review. Dr. Seggelmann is a respected academic and computer science researcher, and there is no possible way to suggest either malice or stupidity here. He did not actually have commit rights to OpenSSL, so he submitted the code to the project members who do have those rights, and they reviewed it. Seeing nothing wrong with the code, and verifying that it did what it said it would do (i.e. implement a Heartbeat) the code was put into production in early 2012.

The problem was discovered by Google researchers and by a Finnish company, Codenomicon, at about the same time, and it was made public in April 2014. There is some suggestion that there was talk from one of the Google people that may have pointed the Codenomicon in the right direction, but perhaps it was simply independent discovery. These things happen. But as Steve Marquess of OpenSSL Foundation

said "The mystery is not that a few overworked volunteers missed the bug; the mystery is why it hasn't happened more often."

TRUECRYPT

The other event I want to talk about is the TrueCrypt audit, which released preliminary results recently. As you may recall, in the wake of the Edward Snowden revelations, there was general anxiety about the security of encryption, and people wanted to know if their encryption had been weakened or a backdoor inserted by the NSA, GCHQ, or other government agencies. In the case of TrueCrypt, you again have an open-source project, with the wrinkle that the developers were deliberately anonymous (and based in Eastern Europe). Pre-Snowden, that might not have aroused too much speculation, but post-Snowden people wanted answers. The TrueCrypt Foundation did the right thing. They raised money (I contributed to the crowd-funded campaign) and enlisted Dr. Matthew Green, a



highly-respected cryptography expert who teaches at Johns Hopkins University, to put together a team to perform an audit of the code. This is a lengthy and difficult task, but the first phase has been completed, and while there are criticisms of certain sloppiness errors, there is no sign of any deliberate errors. You can read a good report on this at novainfosec.com, and that article has a link to the actual report if you want to read it.. This first phase looked at the bootloader and the Windows kernel driver implementations. There is a second phase planned, to go into the cryptography itself, which will use a completely different team of researchers.

So what were the results? Well, TrueCrypt is not perfect, but to expect that would be unrealistic in any case. The audit team did find a certain amount of sloppiness, which probably derives from the fact that the project was done by volunteers and grew organically. But the audit team found no evidence in Phase 1 that there were any deliberate problems or “back doors” in the code. This is good news since this is one of the major open-source programs to

offer serious encryption. If you want to encrypt a directory, a drive, or an entire computer, this will do the job for you, and so far there is no evidence that the encryption is compromised (though there are things they can do to tighten up the code). And of course we should wait for the Phase 2 audit before giving them a clean bill of health.

LESSONS LEARNED

These programs are important to the Internet, so where was the support? This gets at a fundamental problem of companies treating Open Source like it is a free lunch. It is not, for, as you should know, There Ain't No Such Thing As A Free Lunch (TANSTAAFL). Open source is really just a different model for developing and supporting software, one that relies on participation by all of the interested parties. If all of these companies were relying on OpenSSL, for instance, where was their participation? After the fact, it looks like many of them woke up. The Linux Foundation has put together a consortium of major companies. To quote from an Ars Technica article

(<http://arstechnica.com/information-technology/2014/04/tech-giants-chastened-by-heartbleed-finally-agree-to-fund-openssl/>) on the subject “Amazon Web Services, Cisco, Dell, Facebook, Fujitsu, Google, IBM, Intel, Microsoft, NetApp, Qualcomm, Rackspace, and VMware have all pledged to commit at least \$100,000 a year for at least three years to the “Core Infrastructure Initiative,” Linux Foundation Executive Director Jim Zemlin told Ars.” This initiative will be aimed at more than just OpenSSL, but that is good. It means that these companies are taking seriously their responsibility to support the code they rely on. This is in great contrast to the somewhat ridiculous move by Theo de Raadt to create a fork called LibreSSL. This sounds more like ego than a constructive move. I would stick with OpenSSL and give LibreSSL a pass until such time as they can show a long track record of success. A good general rule in security is that new code is more dangerous than code that has been around for a long time.

Security is hard, and is a different skill set than most development. Dr. Seggelmann is a smart guy who was trying to

implement a requirement in an RFC. His code did in fact do that. It was reviewed by someone else on the OpenSSL team, and they did not see any problems with it and put it into production. It sat there for two years before someone noticed a potential problem. The reason a number of smart people missed this is that it takes a different skill set to do security. In hindsight, it is easy to say they should have brought in a specialist, and I think the Core Infrastructure Initiative will help address this.

Bugs are not shallow if the eyeballs are not there. Both TrueCrypt and OpenSSL had small groups of developers with limited resources. Everyone else just assumed that the code was fine, and never tried to look at it. And given that Security requires a specialized skill set, just adding eyeballs is not enough, they need to be the right kind of eyeballs. A question this raises in my mind is about the governance of critical open source projects. Perhaps we need a little more structure to the process to avoid these kinds of problems.

Fixing this requires money, among other things. One of the key take-aways regarding the OpenSSL

project is that they were on what I called a “shoestring” budget, where on average they received \$2,000 per year in donations. Contrast this with the cost of the TrueCrypt audit, where they appear to have raised about \$60,000 so far, and I doubt that is any too much. They put together a team of professionals who understand the work, and that can go through \$60,000 in no time. I always tell people they need to support Free Software, and that includes financial support. If you are only interested in what you can get for free, you will get these kinds of results because the resources will not be there.

The advantage of Open Source software is not that it is bug-free. No software of any kind is bug-free. We make a grave mistake to think so. And it probably is not even correct to think that Open Source has fewer bugs. As we have seen, the weakness of the “many eyeballs – shallow bugs” theory is that, for many Open Source projects, even critical ones, there simply are not that many eyeballs, and often the ones that are there may not be the ones we need to detect subtle problems such as security issues. That does not imply

the opposite, however. The idea that Open Source has issues does not mean that Proprietary Software does any better, as the recent IE bug illustrates (as I write this, people are being advised to stop using IE altogether because of a fundamental security issue. Look up “Operation Clandestine Fox” if you want more details.) The superiority of Open Source is principally that issues generally are addressed quickly. Patches for the Heartbleed bug started to roll out within hours of the disclosure. Patches for the IE bug will at best show up in the next round of Microsoft patches, which could mean waiting a month. Furthermore, with Open Source the whole code is on display, so the quality of our information is much better. With proprietary software the code is never available, the information about the bug tends to be sketchy at best, and in some cases companies will try to keep any information from going out because it could have an adverse effect on their bottom line.

In the case of OpenSSL, Simon Phipps has offered a very interesting article (<http://podcasts.infoworld.com/d/open-source-software/heartbleed->

[postmortem-openssls-license-discouraged-scrutiny-241781?source=rss_security](http://podcasts.infoworld.com/d/open-source-software/heartbleed-postmortem-openssls-license-discouraged-scrutiny-241781?source=rss_security)), based on work of David Wheeler, that points to the license as a source of problems. OpenSSL used a license all of their own which was copy-left, but incompatible with the GPL. And this creates a disincentive for anyone to get involved. He quoted Eben Moglen as saying that the open source license acts as the “constitution of the community” which governs how everyone participates. By having a license that no one else uses, they had the effect of putting in ground rules for participation that no one else understood. The lesson here is that you should not try to re-invent the wheel. There are plenty of good, well-understood, open source licenses out there, and you should use one of them so that the largest number of contributors will be involved. This is one of the reasons that Phipps, Executive Director of OSI, strongly discourages any new license applications. It just isn’t a good idea, and people need to stop this needless proliferation

ADDENDUM

TRUECRYPT STATUS?

June 10th, 2014

by Michael Kennedy

Ironically, an event at the end of May 2014 delivered a further, and still extremely mysterious, security lesson. The TrueCrypt website was suddenly changed:

- It advised users that TrueCrypt (TC) was insecure.
- It recommended users to migrate to BitLocker (a Microsoft product, proprietary, runs on some editions of Vista, Win-7, Win-8, and Win-Servers).
- All the forum’s messages are gone - which has annoyed many.
- And the download links retrieved TC version 7.2 (for Linux, Windows, and Mac OS X platforms), but these builds appear to allow TC users to handle already encrypted TC data – but not to create new TC volumes.

SPECULATION

There has been massive speculation on this development. A

few examples:

- Has the website been hijacked - maybe by some other encryption organisation - and the TC authors/owners not bothered to react?
- Did that recent Audit (as covered by Kevin), or some similar review, detect some weakness, or some backdoor, and have the developers abandoned TC?
- Why is BitLocker, specifically, recommended?
- Was TC too secure, and has the/some government, NSA, etc, tried to kill TC?
- Did the/some government put pressure on the developers (to insert backdoors, etc) - which they resisted?
- Was the/some government or the NSA, etc, behind TC in the first place, and was their cover about to be blown?
- Did the developers just abandon TC? Unhappy?, paid off?, backdoors/hacks revealed?
- This list goes on...

WHAT NOW?

As of June 10th, I don't know who/what to believe. I've used TrueCrypt for some years, on Linux and Windows platforms, and recommended it to clients. From a usage perspective, TC is a super product, cross-platform, and a pleasure to deploy and use. However, until the current TC status is clarified, I'm recommending:

- If you're an existing TC user, and if you're using builds prior to 7.2, hopefully, it's OK to continue to use it?
- If you're on 7.2, or if you're planning to adopt TC, then seek an older TC build, or an alternative product that suits your needs.

Some Comments, References, Alternatives:

- TrueCrypt's own website - <http://truecrypt.sourceforge.net/>
- TC Version 7.1a (all platforms, executables, some sources) - <http://truecrypt.ch/downloads/>
- Ars Technica - <http://arstechnica.com/security/20>

14/05/truecrypt-is-not-secure-official-sourceforge-page-abruptly-warns/

- Bruce Schneier (TrueCrypt WTF) - https://www.schneier.com/blog/archives/2014/05/truecrypt_wtf.html
- Bruce Schneier (Auditing TC, now somewhat out-of-date) - https://www.schneier.com/blog/archives/2014/04/auditing_truecrypt.html
- Steve Gibson's take (GRC) - <https://www.grc.com/misc/truecrypt/truecrypt.htm>
- Slashdot - <http://it.slashdot.org/story/14/05/28/2126249/truecrypt-website-says-to-switch-to-bitlocker>
- Alternatives at Wikipedia (See also 7-Zip, and VeraCrypt, DCrypt, etc, on Sourceforge) - http://en.wikipedia.org/wiki/Comparison_of_disk_encryption_software
- PS: Good summary in WindowsSecrets Newsletter - <http://windowssecrets.com/newsletter/the-life-and-untimely-demise-of-truecrypt/>

The mistake these developers made was in believing that they still “owned” TrueCrypt, and that it was theirs to kill.

Those who believe that there is something suddenly “wrong” with TrueCrypt because its creators have decided they no longer have so much to give are misguided.

Note that once TrueCrypt has been independently audited it will be the only mass storage encryption solution to have been audited. This will likely cement TrueCrypt's position as the top, cross-platform, mass storage encryption tool.

- Steve Gibson
<https://www.grc.com/misc/truecrypt/truecrypt.htm>



On April 17 2014, Canonical released Ubuntu 14.04 LTS. I like the latest in software, but at the same time I like stability, which is what Ubuntu's Long Term Support cycle is all about. So, after two years of using Ubuntu 12.04 LTS, it was time for me to upgrade, and so I did. For many of us who have been using Steam to play video games, a new release means having to re-download and re-install a ton of games; which can take countless hours, possibly even days. No one wants to spend what seems like an eternity re-downloading and re-installing every game we already own before being able to pick up playing where we left off. However, fortunately for us, there is a better and faster way to transfer ALL of our installed Steam games from one hard drive to another.

The basic concept of how to transfer your Steam game library from one hard drive (or partition) to another is relatively basic to understand. In a nutshell, all you've got to do is copy your library from your old location to your new one,

and re-start the Steam client, which will automatically recognize your games and you can get back to playing right away.

Doing it is a different story. Although not terribly complicated, you must follow every step exactly and in the correct order. So let me explain how it's done.

- First thing's first, you MUST install Steam on your new system (and if you never had the Steam client installed before, then, obviously, this guide is pointless for you).
- Having installed Steam on your new system, you MUST start Steam so that it can create all of the

folders & files it needs to operate. You also HAVE to sign in with the Steam account that you've been using.

- Having started Steam & having signed in, you MUST now Exit Steam by going to the upper left and then selecting Steam>Exit. Simply clicking on the Close (X) button on the top right will not work because doing this doesn't really close Steam; it only closes Steam's user interface but keeps the software running in the background.
- Open the "home" folder in your old system and after it's opened press Ctrl+H on your keyboard in

Moving Steam

order to see your old "home" folder's hidden files & folders. I find it easiest to hold Ctrl down while I press H, in other words both must be pressed at the same time. Repeat this step for your new "home" folder.

- Now go to the following directory in the old "home":

`.local/share/steam` and copy it.

- Paste the copied folder to your new `/home/username/.local/share/steam` and wait until the entire contents of `/home/username/.local/share/steam`



m has been copied to the new location. You're almost done!

- Re-start Steam and wait for the Steam client to recognize your newly copied old games.
- Start playing games in your new system!

There's also a way to do everything I just described from the terminal, but I found it easier to do it using the graphic-user-interface. Not only that, but I imagine that if you would rather use the terminal, you may not even need this guide. As a reference, I used the following web page, especially Damienov's answer: <http://steamcommunity.com/app/21410/discussions/0/882965239721861812/?l=polish#p2>

I originally transferred my games from my old Ubuntu installation to my new one, but to be certain that the procedure I described works as efficiently as I claim, I verified it by following the steps I've suggested with a separate installation of Linux Mint. The computer I used was my custom made desktop PC consisting of an AMD FX-6100 3.3GHz CPU, an Asus M5A97-EVO

motherboard, a Sapphire Radeon HD 5770 graphics card, 8GB of Kingston Hyper X RAM, and a 1TB Seagate Barracuda hard drive. The software used was Ubuntu 14.04 LTS with Unity and AMD 13.35 proprietary graphic drivers.

COMPETITION:

On a final note, I'd like to congratulate **David** and **Earl** on correctly answering last month's question for which they each got a *Humble Indie Bundle*. This month's question is the following:

What do you need to do to be able to see your "home" directory's hidden folders & files?

Send an email to 7bluehand@gmail.com with the answer.

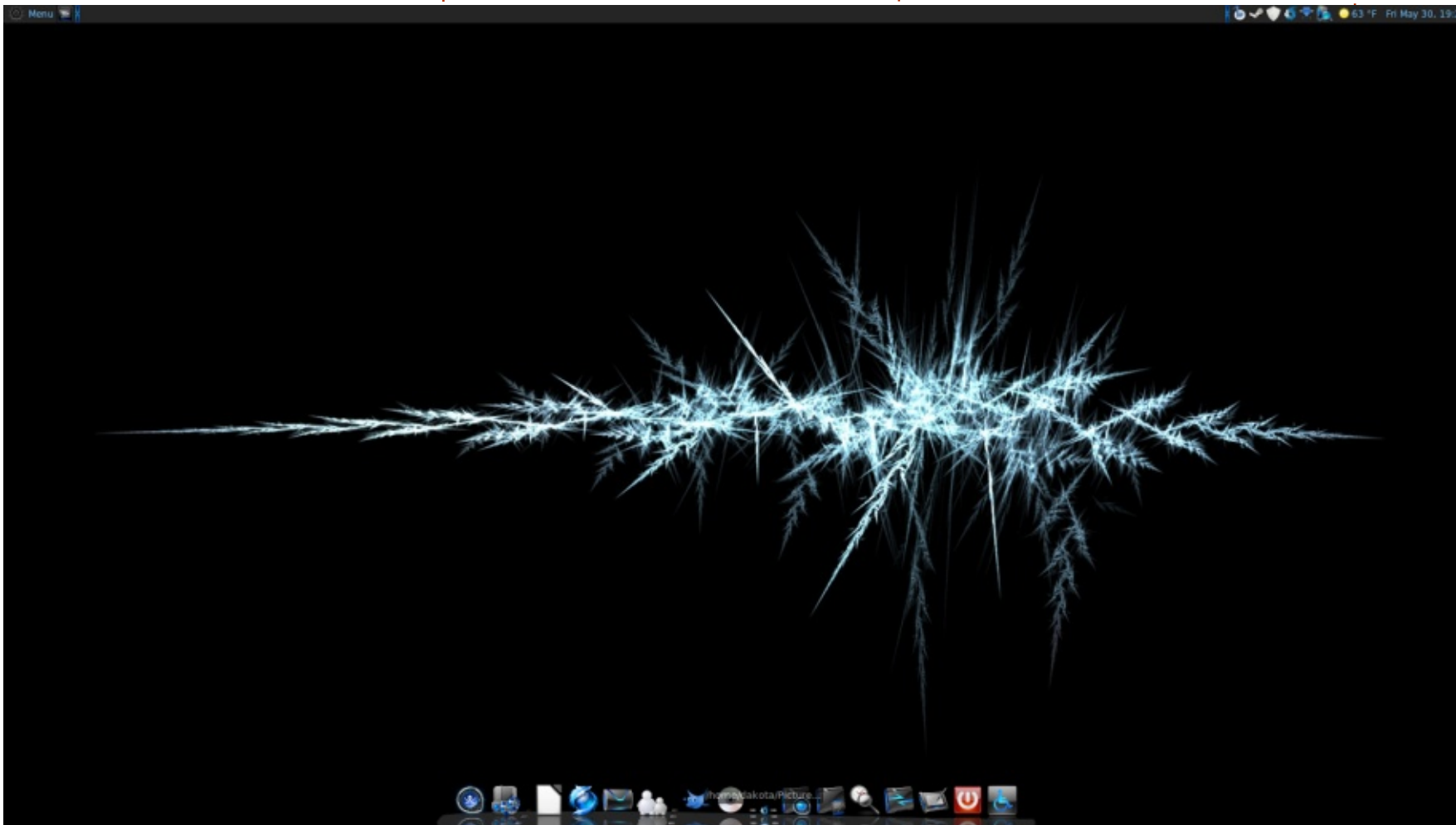


Oscar graduated from CSUN, is a Music Director/Teacher, beta tester, Wikipedia editor, and Ubuntu Forums contributor. You can contact him via: www.gplus.to/7bluehand or email: 7bluehand@gmail.com



MY DESKTOP

Your chance to show the world your desktop or PC. Email your screenshots and photos to: misc@fullcirclemagazine.org and include a brief paragraph about your desktop, your PC's specs and any other interesting tidbits about your setup.



It didn't want to change the window title bars in cinnamon, and I hadn't had a chance to try Mate yet, so out the door it went.

I found the background picture by googling "Dark Wallpaper" in Google Images. It can be found here:
<http://wall.alphacoders.com/big.php?i=70225>

Then I just installed them using the Appearance app in the Control Center. I did have to get a little sneaky, and use "sudo mv" to put the background image into the folder required by the theme manager. It

is located here:

`/usr/share/backgrounds/linuxmint-qiana`

David Harbour

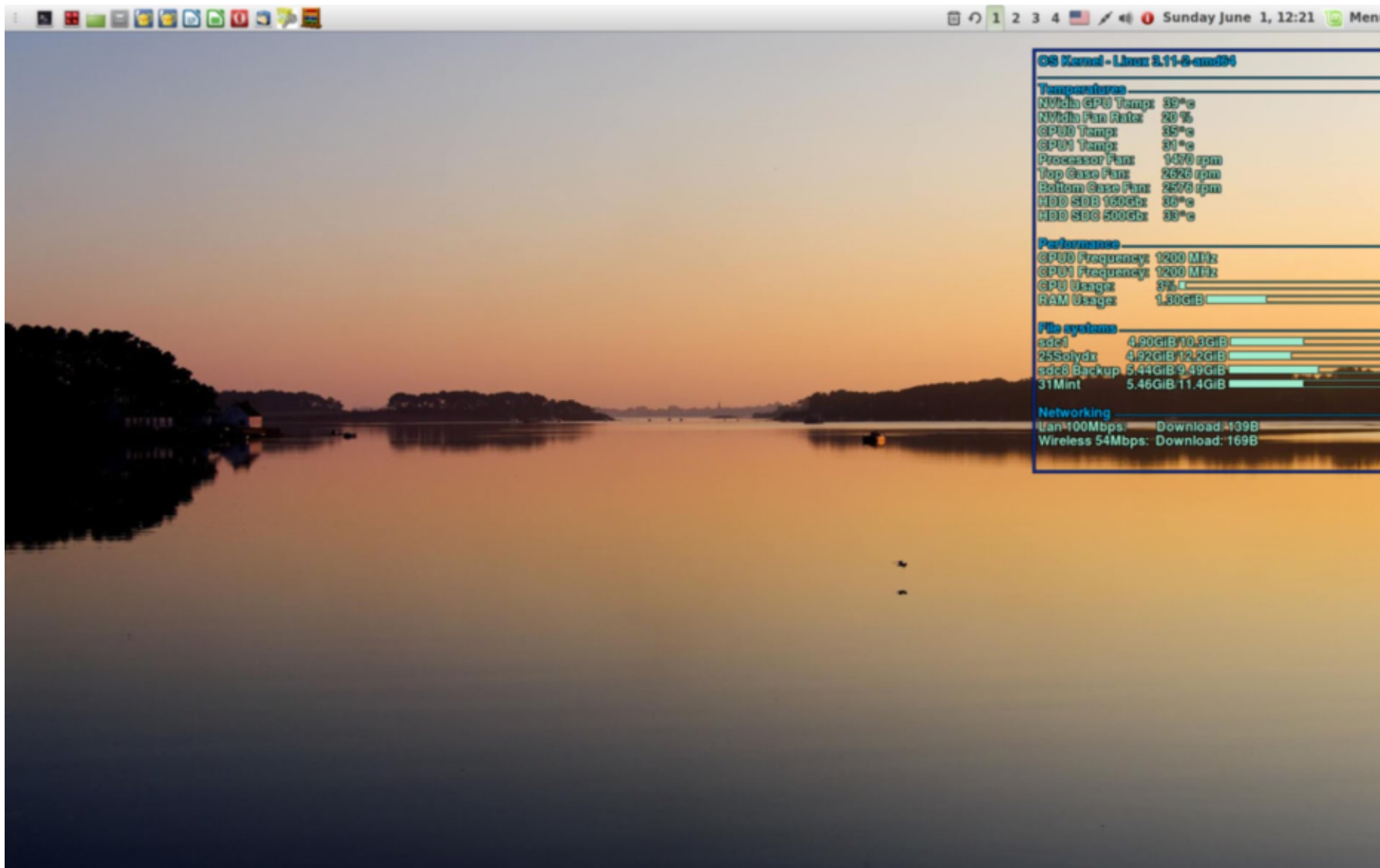
I tried each of the {K,L,X,}ubuntu 14.04 distros, but, for some reason, something wasn't working in each. Linux Mint had just released 17 so I tried Cinnamon. Everything worked out of the box. I

knew what I wanted my desktop to look like so I started trying to figure out how to change it and I found this theme on gnome-look.org:
<http://gnome-look.org/content/show.php/Best-Of-Darkblue?content=164207>

<http://gnome-look.org/content/show.php/Best-Of-Dark?content=164206> and
<http://gnome-look.org/content/show.php/Best-Of-Darkblue?content=164207>

I needed both installed.





My desktop must be clean and restful as I spend many hours on the computer. I am running a basic dual-core computer with Gigabyte M/b and 4GB RAM. I run Linux 16 or LMDE from an SSD drive with 3 spare older drives for

testing alternative Linux distributions. The monitor is Samsung 22" wide (1650 x 1050). I use Cinnamon desktop with MintX Theme. I change the wallpaper frequently; this one is "Morning Lake." I use conky to provide

comprehensive system data without intruding too much on the peaceful scene.

Michael Davies



HOW TO CONTRIBUTE

FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your opinions, desktops, stories, how-to's, reviews, and anything else you want to tell your fellow *buntu users. Send your articles to: articles@fullcirclemagazine.org

We are always looking for new articles to include in Full Circle. For help and advice please see the **Official Full Circle Style Guide**: <http://url.fullcirclemagazine.org/75d471>

Send your **comments** or Linux experiences to: letters@fullcirclemagazine.org
Hardware/software **reviews** should be sent to: reviews@fullcirclemagazine.org
Questions for Q&A should go to: questions@fullcirclemagazine.org
Desktop screens should be emailed to: misc@fullcirclemagazine.org
... or you can visit our **forum** via: fullcirclemagazine.org

FCM#87

Deadline:

Sunday 06th July. 2014.

Release:

Friday 25th July. 2014.



Full Circle Team



Editor - Ronnie Tucker

ronnie@fullcirclemagazine.org

Webmaster - Rob Kerfia

admin@fullcirclemagazine.org

Podcast - Les Pounder & Co.

podcast@fullcirclemagazine.org

Editing & Proofreading

Mike Kennedy, Gord Campbell, Robert Orsino, Josh Hertel, Bert Jerred, Jim Dyer and Emily Gonyer

Our thanks go to Canonical, the many translation teams around the world and **Thorsten Wilms** for the FCM logo.

Getting Full Circle Magazine:



EPUB Format - Recent editions of Full Circle have a link to the epub file on the downloads page. If you have any problems with the epub file, you can drop an email to: mobile@fullcirclemagazine.org



Issuu - You can read Full Circle online via Issuu: <http://issuu.com/fullcirclemagazine>. Please share and rate FCM as it helps to spread the word about FCM and Ubuntu Linux.

